

FMT character Notebook

[Code ▼](#)

This is an R Markdown (<http://rmarkdown.rstudio.com>) Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Cmd+Shift+Enter*.

[Hide](#)

```
library(ggplot2)
# library(export)
library(cluster)
library(clusterSim)
library(ade4)
library(pROC)
library(reshape2)
library(ggpubr)
library(coin)
library(gridExtra)
library(ape)
```

[Hide](#)

```
setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
getwd()
```

```
[1] "C:/work/fmt_enterotype/a_microbiome/analysis"
```

[Hide](#)

```
source("pre_processing.R")
```

```
[1] 0.01
[1] 0.01
[1] 0.01
[1] 0.01
```

[Hide](#)

```
rownames(arare) <- arare[,1]
alph_dat <- arare[unique(c(meta_fil_config_entro$SRA_Sample, meta_fil_config_entro$Donor_sra, m
eta_fil_config_entro$Previous_sra)),]
length(unique(meta_fil_config_entro$Previous_sra))
```

```
[1] 322
```

[Hide](#)

```

library(ggbeeswarm)
library(ggpubr)
donor_before_after_color <- c("#9F452A", "#4E86C6", "#235E27")
library(ggplot2)
#ERS1600370
##sex alpha boxplot
as.data.frame(unique(cbind(meta_fil_config_entro$Previous_sra, meta_fil_config_entro$PRJ, meta_
fil_config_entro$Sex, alph_dat[meta_fil_config_entro$Previous_sra, "shannon"])))>sex_alpha
colnames(sex_alpha) = c("after", 'prj', 'sex', 'alpha')
sex_alpha$alpha <- as.numeric(as.character(sex_alpha$alpha))
sex_alpha[sex_alpha$sex %in% c('F', 'M'),]->sex_alpha

library(coin)
wilcox_test(alpha ~ sex | prj, data = sex_alpha)

```

Asymptotic Wilcoxon-Mann-Whitney Test

data: alpha by sex (F, M)
 stratified by prj
 Z = 1.238, p-value = 0.2157
 alternative hypothesis: true mu is not equal to 0

Hide

```
wilcox_test(alpha ~ sex , data = sex_alpha)
```

Asymptotic Wilcoxon-Mann-Whitney Test

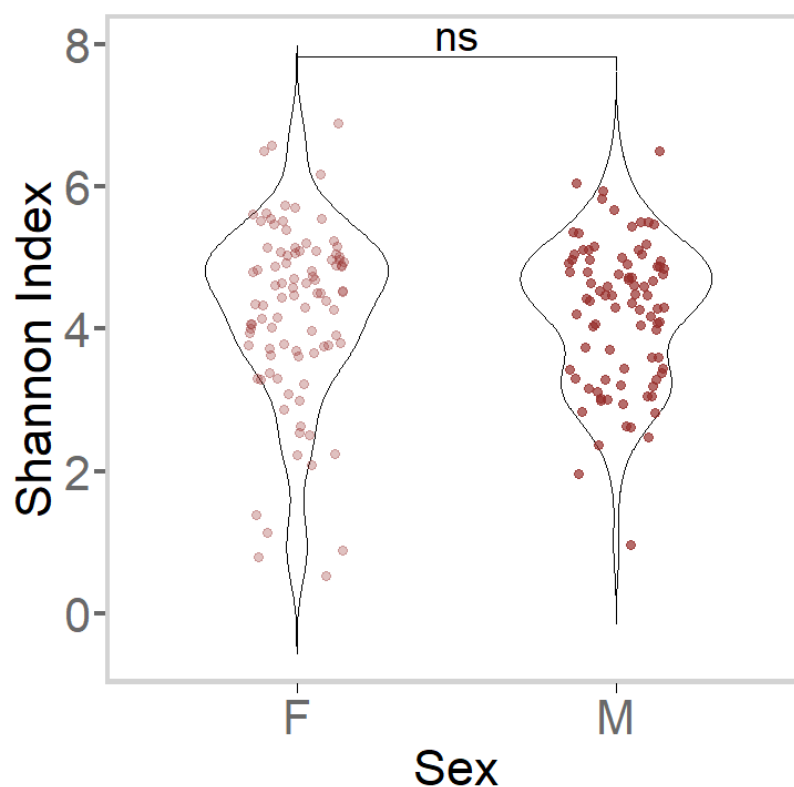
data: alpha by sex (F, M)
 Z = 0.66702, p-value = 0.5048
 alternative hypothesis: true mu is not equal to 0

Hide

#Z = -1.2026, p-value = 0.2291

```
ggviolin(sex_alpha, x='sex', y='alpha', outlier = 0, na.rm = T, width = 0.6)+
  # geom_boxplot(aes(, ), alpha=0.3,)+
  geom_jitter(aes(x=sex, y=alpha, col=sex, alpha=sex), size=3, width = 0.15)+
  # theme_bw()+theme(legend.position = "none")+xlab(label = '')+ylab("Shannon")+labs(title='Sex')
  # theme( axis.title=element_text(size=12), aspect.ratio=0.68)+
  # geom_quasirandom(aes(color=don_entro), method='smiley', size=3)+
  # scale_y_continuous(expand = c(0, 0.1))+
  # geom_hline(yintercept = 0, linetype=2)+
  # scale_alpha_continuous( range = c(0.6, 1))+
  stat_compare_means(comparisons = list(c(unique(sex_alpha$sex))), method = 'wilcox.test', label = "p.signif", label.y = 7.82, size=10)+
  labs(x= c('Sex'), y=c('Shannon Index'), title = c(''))+
  scale_alpha_discrete(range=c(0.3, 0.7))+
  scale_color_manual(name="FMT", values=c("#962E2B", "#962E2B", "#4E86C6", "#E7A600", "#4D9127", "#90908D",
    donor_before_after_color, "#962E2B", "#4E86C6", "#4D9127", "#90908D", 'lightgrey'))+
  # theme(text=element_text(family = "sans", size=28), plot.title = element_text(size=24, hjust = 0.5), axis.title.x = element_text(size=24), axis.text.x = element_text(size=26))+
  theme(text=element_text(family = "sans", size=32), plot.title = element_text(size=34, hjust = 0.5), axis.text = element_text(size=32, color = 'dimgray'), axis.title.x = element_text(size=34), axis.title.y = element_text(size=34), axis.ticks.y = element_line(size=1.5, color = 'dimgray'), axis.ticks.length = unit(7, "pt"))+
  theme(aspect.ratio = 0.95, legend.background=element_blank(), legend.position=c(1.75, 0.6),
    ,panel.background = element_rect(fill = NA, colour = "lightgrey", size = 3)
    ,axis.line=element_line(colour=NA)
    ,legend.key = element_rect(fill = NA, color = NA))+
  guides(colour = guide_legend(override.aes = list(size=5)))
```

Using alpha for a discrete variable is not advised.



[Hide](#)

```
figi = 1
ggsave(paste("./figure2/fig2_", figi, ".pdf", sep = ''), device = "pdf", useDingbats=FALSE)
```

Saving 12.9 x 8 in image

[Hide](#)

```
figi = figi + 1
```

[Hide](#)

```
##time alpha boxplot
as.data.frame(cbind(meta_fil_config_entro$SRA_Sample, meta_fil_config_entro$PRJ, meta_fil_config_entro$Time, alph_dat[meta_fil_config_entro$SRA_Sample, "shannon"]))->time_alpha
colnames(time_alpha) = c("after", 'prj', 'time', 'alpha')
time_alpha$alpha <- as.numeric(as.character(time_alpha$alpha))
time_alpha$time<-as.numeric(as.character(time_alpha$time))
time_alpha[is.na(time_alpha$time),]
```

0 rows

[Hide](#)

```
#hist(time_alpha[time_alpha$time < 60, 'time'])
time_alpha$type <- 0

time_alpha[time_alpha$time < 7, "type"] = '<7d'
time_alpha[7 <= time_alpha$time & time_alpha$time < 31, "type"] = '<1m'
time_alpha[31 <= time_alpha$time & time_alpha$time < 93, "type"] = '<3m'
time_alpha[93 <= time_alpha$time, "type"] = '>3m'

time_alpha$type <- as.factor(time_alpha$type)
library(coin)
wilcox_test(alpha ~ type | prj, data = time_alpha[time_alpha$type %in% c('<7d', '<1m'),])
```

Asymptotic Wilcoxon-Mann-Whitney Test

```
data: alpha by type (<1m, <7d)
      stratified by prj
Z = 3.9526, p-value = 7.731e-05
alternative hypothesis: true mu is not equal to 0
```

[Hide](#)

```

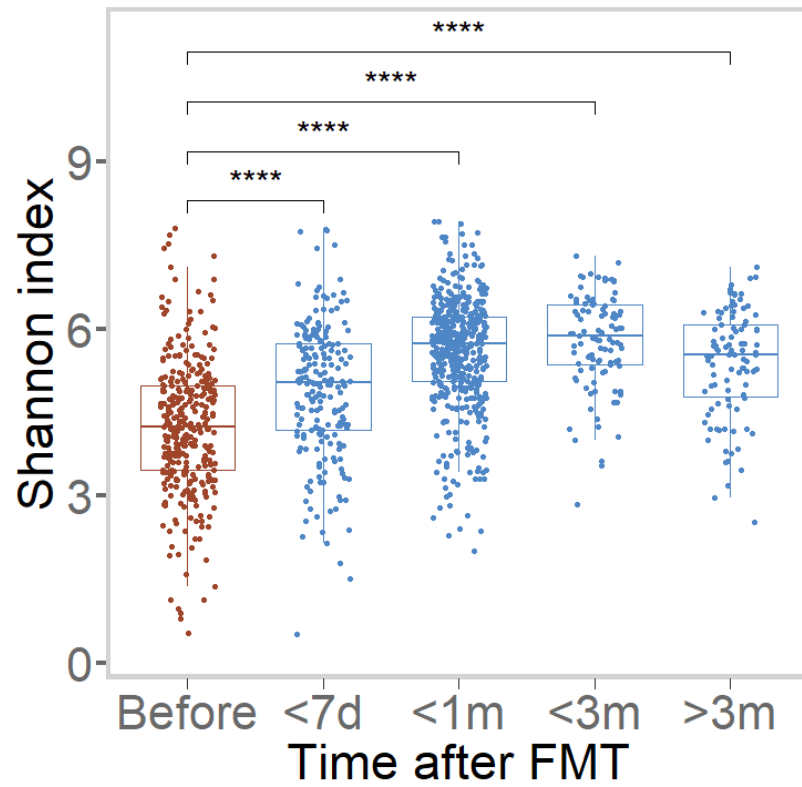
#Z = -3.7094, p-value = 0.0002078
#Z = -4.4906, p-value = 7.101e-06
#Z = -3.9396, p-value = 8.161e-05

##add previous
as.data.frame(cbind(meta_fil_config_entro$Previous_sra, meta_fil_config_entro$PRJ, 'Before', alpha_dat[meta_fil_config_entro$Previous_sra, "shannon"]))->pre_alpha
colnames(pre_alpha) = c("after", 'prj', 'time', 'alpha')
unique(pre_alpha)->pre_alpha
pre_alpha$type <- 'Before'
rbind(time_alpha, pre_alpha) -> ptime_alpha

ptime_alpha$type = factor(ptime_alpha$type, levels=c('Before', '<7d', '<1m', '<3m', '>3m'))
ptime_alpha$alpha <- as.numeric(as.character(ptime_alpha$alpha))
#
# ggplot(ptime_alpha)+geom_boxplot(aes(type, alpha), alpha=0.3,outlier.alpha = 0, na.rm = T)+geom_jitter(aes(x=type, y=alpha, col=type))+
#   theme_bw()+theme(legend.position = "none")+xlab(label = '')+ylab("Shannon")+labs(title='Time')
#   theme(axis.title=element_text(size=14), axis.text=element_text(size=12), aspect.ratio=0.68)

library(ggpubr)
ggboxplot(ptime_alpha, x="type", y="alpha", col="type", add='jitter',
          palette = c("#9F452A", "#4E86C6", "#4E86C6", "#4E86C6", "#4E86C6"))+
  scale_y_continuous(expand = c(0, 0.8))+
  # geom_quasirandom(aes(color=type), method='smiley', size=0.3)+
  stat_compare_means(comparisons = list(c('Before', '<7d'), c('Before', '<1m'), c('<3m', 'Before'), c('>3m', 'Before')), method = 'wilcox.test', label = "p.signif", size=8)+
  theme_classic()+theme(legend.position = "none")+xlab(label = 'Time after FMT')+ylab("Shannon index")+labs(title='')+
  theme(text=element_text(family="sans", size=32), plot.title = element_text(size=34, hjust = 0.5), axis.text = element_text(size=32, color='dimgray'), axis.title.x = element_text(size=34), axis.title.y = element_text(size=34), axis.ticks.y = element_line(size=1.5, color='dimgray'), axis.ticks.length = unit(7, "pt"))+
  theme(aspect.ratio = 0.95, legend.background=element_blank(), legend.position=c(1.75, 0.6),
        ,panel.background = element_rect(fill = NA, colour = "lightgrey", size = 3)
        ,axis.line=element_line(colour="lightgrey")
        ,legend.key = element_rect(fill = NA, color = NA))+
  guides(colour = guide_legend(override.aes = list(size=5)))

```



Hide

```
ggsave(paste("./figure2/fig2_s_", figi, ".pdf", sep = ''), device = "pdf", useDingbats=FALSE)
```

Saving 12.9 x 8 in image

Hide

```
figi = figi + 1
```

Hide

```

###Before enterotypes alpha boxplot
##before_entro_prj      don_entro_prj

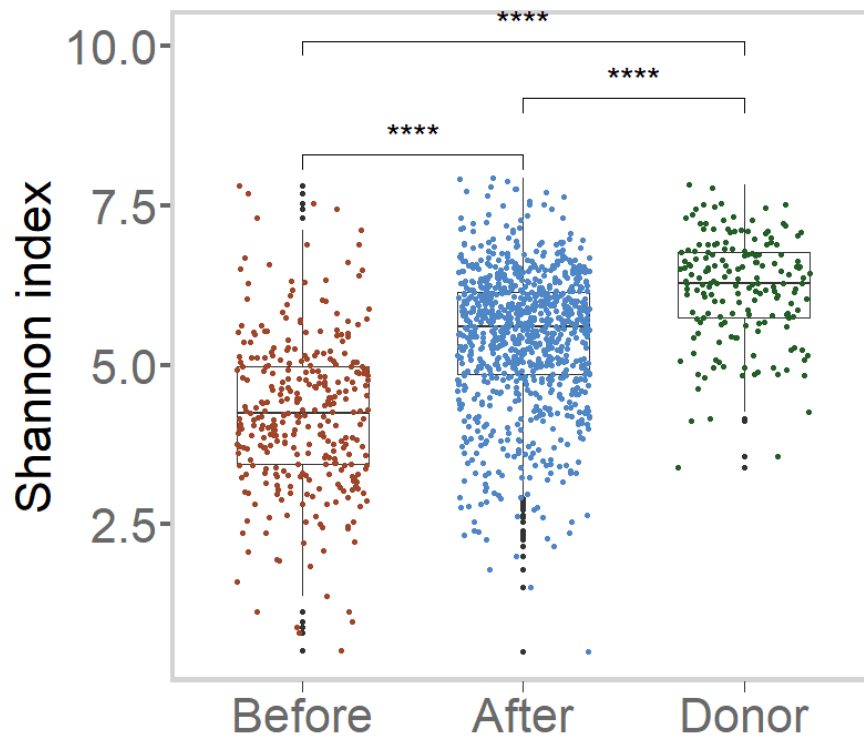
previous_alpha <- unique(as.data.frame(cbind(meta_fil_config_entro$Previous_sra, meta_fil_config_entro$PRJ, 'Before', as.character(meta_fil_config_entro$pre_entro), as.character(meta_fil_config_entro$Dieasel), alph_dat[as.character(meta_fil_config_entro$Previous_sra), "shannon"])))
after_alpha <- unique(as.data.frame(cbind(meta_fil_config_entro$SRA_Sample, meta_fil_config_entro$PRJ, 'After', as.character(meta_fil_config_entro$pre_entro), as.character(meta_fil_config_entro$Dieasel), alph_dat[as.character(meta_fil_config_entro$SRA_Sample), "shannon"])))
donor_alpha <- unique(as.data.frame(cbind(meta_fil_config_entro$Donor_sra, meta_fil_config_entro$PRJ, 'Donor', 'Donor', as.character(meta_fil_config_entro$Dieasel), alph_dat[as.character(meta_fil_config_entro$Donor_sra), "shannon"])))

rbind(previous_alpha, after_alpha, donor_alpha) -> fmt_alpha
colnames(fmt_alpha) = c("id", 'prj', 'FMT', 'Enterotypes', 'Disease', 'shannon')

fmt_alpha$FMT = factor(fmt_alpha$FMT, levels=c('Before', 'After', 'Donor'))
fmt_alpha$shannon <- as.numeric(as.character(fmt_alpha$shannon))

ggboxplot(fmt_alpha, x='FMT', y='shannon', color=NA)+geom_boxplot(aes(FMT, shannon), width=0.6)
+
  geom_jitter(aes(x=FMT, y=shannon,color=FMT), width=0.3)+
  stat_compare_means(comparisons = list(c('Before', 'After'), c('Donor', 'After'), c('Donor', 'Before')), method = 'wilcox.test', label = "p.signif", size=8)+
  scale_colour_manual(values = donor_before_after_color)+
  theme_classic()+theme(legend.position = "none")+xlab(label = '')+ylab("Shannon index")+labs(title='')+
  theme(text=element_text(family = "sans", size=32), plot.title = element_text(size=34, hjust = 0.5), axis.text = element_text(size=32, color = 'dimgray'), axis.title.x = element_text(size=34), axis.title.y = element_text(size=34), axis.ticks.y = element_line(size=1.5, color = 'dimgray'), axis.ticks.length = unit(7, "pt"))+
  theme(aspect.ratio = 0.95, legend.background=element_blank(), legend.position=c(1.75, 0.6)
    ,panel.background = element_rect(fill = NA, colour = "lightgrey", size = 3)
    ,axis.line=element_line(colour="lightgrey")
    ,legend.key = element_rect(fill = NA, color = NA))+
  guides(colour = guide_legend(override.aes = list(size=5)))

```



Hide

```
ggsave(paste("./figure2/fig2_", figi, ".pdf", sep = ''), device = "pdf", useDingbats=FALSE)
```

Saving 12.9 x 8 in image

Hide

```
figi = figi + 1
```

Hide

```
previous_response_alpha <- unique(as.data.frame(cbind(meta_fil_config_entro$Previous_sra, meta_fil_config_entro$PRJ, 'Before', as.character(meta_fil_config_entro$pre_entro), as.character(meta_fil_config_entro$postfmt_symptoms), (alph_dat[as.character(meta_fil_config_entro$Previous_sra), "shannon"])), stringsAsFactors = F))
colnames(previous_response_alpha) = c("id", 'prj', 'FMT', 'Enterotypes', 'post', 'shannon')

library(coin)
previous_response_alpha_f <- previous_response_alpha[previous_response_alpha$post %in% c("response", "failure"),]
# previous_response_alpha_f$post <- ifelse(previous_response_alpha_f$post=='failure', 1, 0)
previous_response_alpha_f$post <- factor(previous_response_alpha_f$post, levels = c('response', 'failure'))
previous_response_alpha_f$prj <- factor(previous_response_alpha_f$prj)
previous_response_alpha_f$shannon <- as.numeric(previous_response_alpha_f$shannon)
wilcox_test(shannon ~ post | prj, previous_response_alpha_f)
```

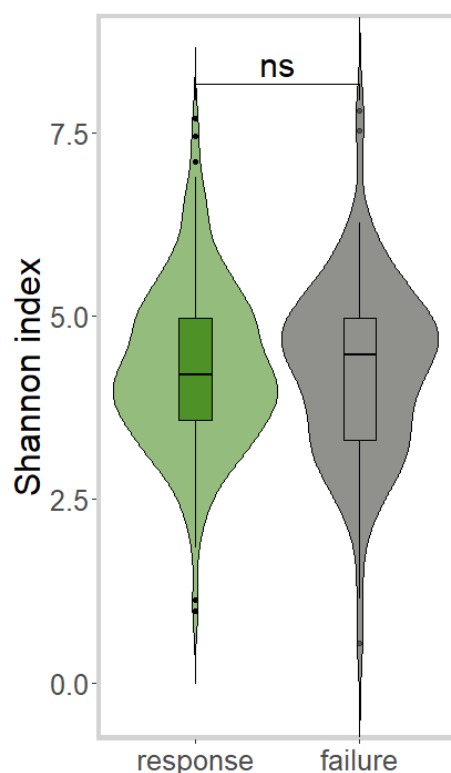

Asymptotic Wilcoxon-Mann-Whitney Test

```
data: shannon by post (response, failure)
      stratified by prj
Z = 1.0613, p-value = 0.2885
alternative hypothesis: true mu is not equal to 0
```

Hide

```
pp_alpha<-ggviolin(previous_response_alpha_f, x='post', y='shannon', fill='post', alpha = 'post',
  add = "boxplot", add.params = list(alpha=c(1, 0.4)), palette = c("#4D9127", "#90908D"))+
  scale_y_continuous(expand = c(0, 0.04))+
  scale_alpha_discrete( range = c(0.6, 1))+
  stat_compare_means(comparisons = list(c('response', 'failure')), method = 'wilcox.test', label = "p.signif", size=8)+
  theme_classic()+theme(legend.position = "right")+xlab(label = '')+ylab("Shannon index")+labs(
  (title='')+ #Donor_receptor distance before FMT
  theme(text=element_text(family = "sans", size=24), plot.title = element_text(size=24, hjust = 0.5),
  axis.title.x = element_text(size=21, vjust = -0.5, hjust = 0.71, color = 'dimgray'))+
  theme(aspect.ratio = 2, legend.background=element_blank(), legend.position=c(5, 0.6)
  ,panel.background = element_rect(fill = NA, colour = "lightgrey", size = 3)
  ,axis.line=element_line(colour="lightgrey")
  ,legend.key = element_rect(fill = NA, color = NA))+
  guides(colour = guide_legend(override.aes = list(size=5)));pp_alpha
```

Using alpha for a discrete variable is not advised.



Hide

```
ggsave(paste("./figure2/fig2_s_", figi, ".pdf", sep = ''), device = "pdf", useDingbats=FALSE)
```

Saving 12.9 x 8 in image

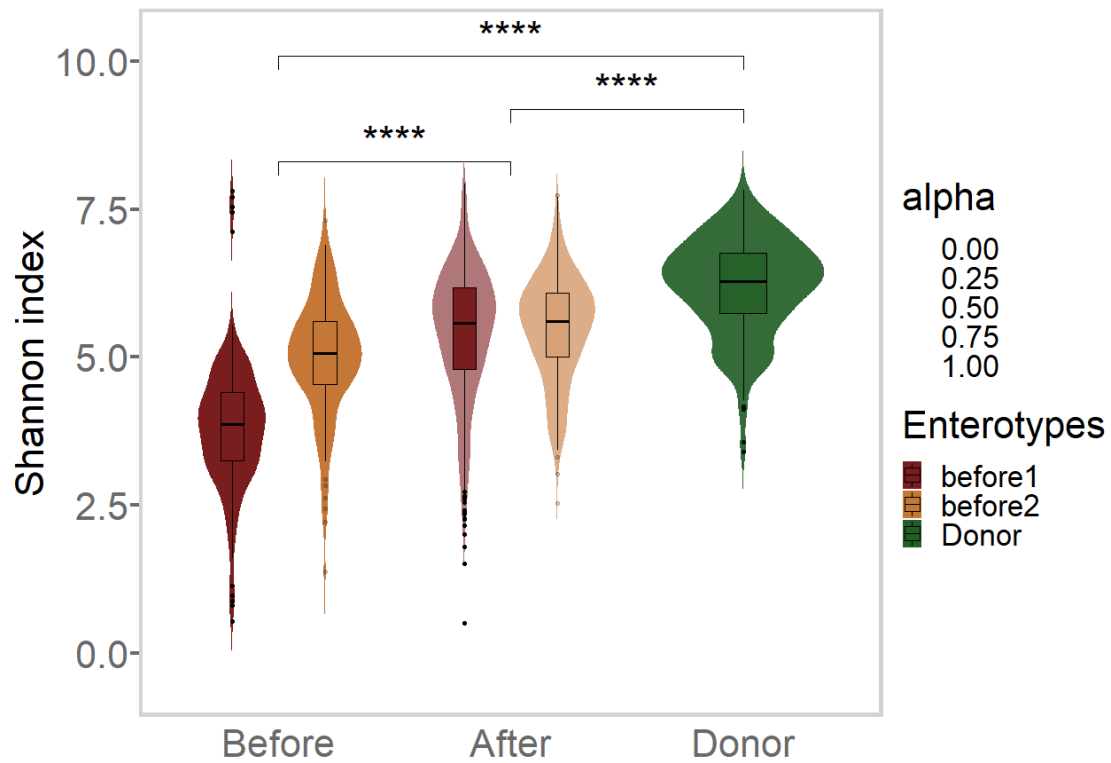
Hide

```
figi = figi + 1
```

Hide

```
library(ggpubr)
fmt_alpha$Enterotypes = factor(fmt_alpha$Enterotypes, levels=c('before1', 'before2', 'Donor'))
fmt_alpha$alpha = as.numeric((ifelse(fmt_alpha$FMT == 'After', 0, ifelse(fmt_alpha$FMT == 'Donor', 0.8, 1))))

ggviolin(fmt_alpha, x="FMT", y="shannon", fill = "Enterotypes", alpha = 'alpha', color='NA', add = "boxplot",
  add.params = list(alpha=c(1, 1, 0.2, 0.2, 0.8), size=0.7, color='black'), palette = c("#7A1D1E", "#C47737", "#235E27"), width = 0.7)+ #
  # geom_boxplot(aes(fill = Enterotypes), width = 0.3, , outlier.size = 0, outlier.color = NA)+
  stat_compare_means(comparisons = list(c('Before', 'After'), c('After', 'Donor'), c('Before', 'Donor')), method = 'wilcox.test', label = "p.signif", size=12)+
  theme_classic()+theme(legend.position = "right")+xlab(label = '')+ylab("Shannon index")+labs(title='')+#Alpha diversity improvement in recipients
  # theme(axis.title=element_text(size=14), axis.text=element_text(size=12), aspect.ratio=0.95,
  # legend.direction = 'horizontal', legend.position = c(.75, .15), legend.background=element_rect(fill = NA)
  # ,panel.background = element_rect(fill = NA, colour = "lightgrey", size = 3)
  # ,axis.line=element_line(colour="lightgrey"))
  scale_alpha_continuous(range = c(0.6, 1))+
  scale_y_continuous(expand = c(0, 0.8))+
  # theme(text=element_text(family = "sans", size=28), plot.title = element_text(size=28, hjust = 0.5), axis.text = element_text(size=24, color = 'dimgray'))+
  theme(text=element_text(family = "sans", size=32), plot.title = element_text(size=34, hjust = 0.5), axis.text = element_text(size=32, color = 'dimgray'), axis.title.x = element_text(size=34), axis.title.y = element_text(size=34), axis.ticks.y = element_line(size=1.5, color = 'dimgray'), axis.ticks.length = unit(7, "pt"))+
  # scale_alpha_continuous(range = c(1, 1, 0.1, 0.1, 1))+
  theme(aspect.ratio = 0.95, legend.background=element_blank(), legend.position=c(1.75, 0.6)
  ,panel.background = element_rect(fill = NA, colour = "lightgrey", size = 3)
  ,axis.line=element_line(colour=NA), axis.ticks.x = element_blank()
  ,legend.key = element_rect(fill = NA, color = NA))+
  # theme(axis.text.y = element_blank(), axis.ticks.y = element_blank(), axis.title.y = element_blank(),)+
  guides(colour = guide_legend(override.aes = list(size=3)))
```



Hide

```
ggsave(paste("./figure2/2main_", figi, ".pdf", sep = ''), device = "pdf", useDingbats=FALSE)
```

Saving 16.2 x 10 in image

Hide

```
figi = figi + 1

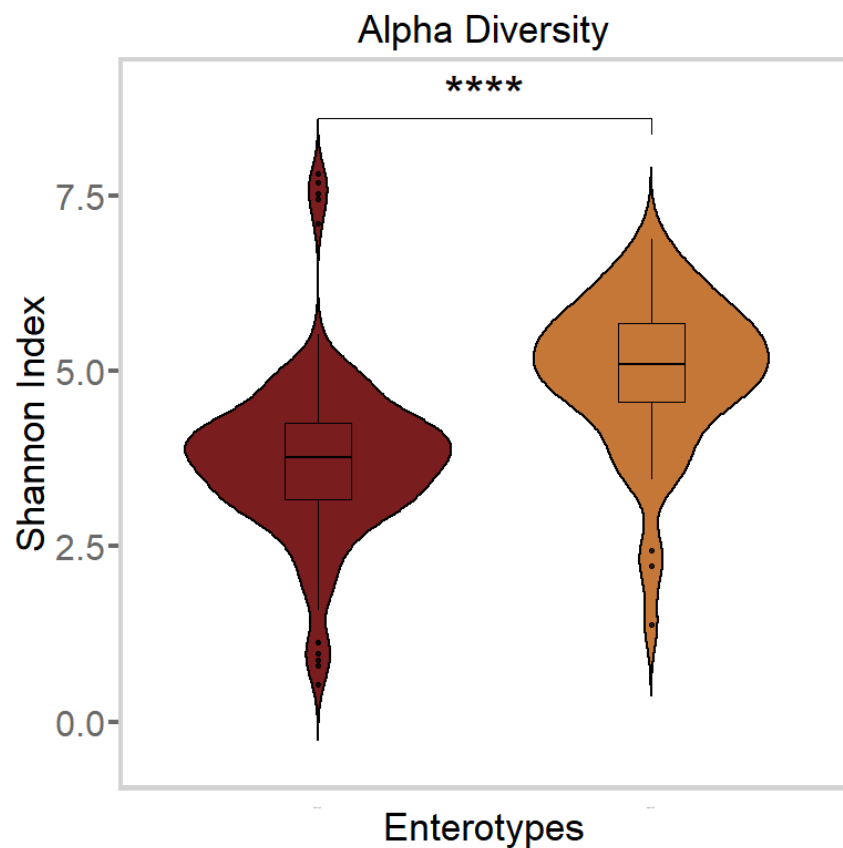
library(coin)
wilcox_test(shannon ~ Enterotypes | prj, data = fmt_alpha[fmt_alpha$Enterotypes %in% c('before1', 'before2'),])
```

Asymptotic Wilcoxon-Mann-Whitney Test

```
data: shannon by Enterotypes (before1, before2)
stratified by prj
Z = -5.0312, p-value = 4.874e-07
alternative hypothesis: true mu is not equal to 0
```

Hide

```
library(ggpubr)
ggviolin(fmt_alpha[fmt_alpha$FMT %in% c('Before') & fmt_alpha$Disease %in% c('CDI')], x="Enterotypes", y="shannon", fill = "Enterotypes", add = "boxplot", alpha=1, add.params = list(alpha=0.8), palette = c("#7A1D1E", "#C47737"), size = 0.8, width = 0.8)+
  stat_compare_means(comparisons = list(c('before1', 'before2')), method = 'wilcox.test', label = "p.signif", label.x = 1.5, label.y = 8.6, size=12)+
  scale_y_continuous(limits = c(-0.5, 9))+
  # yscale("log2", .format = FALSE)+
  theme_classic()+theme(legend.position = "right")+xlab(label = 'Enterotypes')+ylab("Shannon Index")+labs(title='Alpha Diversity')+
  theme(text=element_text(family = "sans", size=24), plot.title = element_text(size=26, hjust = 0.5), axis.text = element_text(size=24, color = 'dimgray'), axis.text.x = element_text(size=0), axis.title.x = element_text(size=26), axis.title.y = element_text(size=26), axis.ticks.y = element_line(size=1.5, color = 'dimgray'), axis.ticks.length = unit(7, "pt"))+
  theme(aspect.ratio = 1., legend.background=element_blank(), legend.position=c(1.75, 0.6),
        ,panel.background = element_rect(fill = NA, colour = "lightgrey", size = 3)
        ,axis.line=element_line(colour=NA)
        ,legend.key = element_rect(fill = NA, color = NA), axis.ticks = element_blank())+
  guides(colour = guide_legend(override.aes = list(size=5)))
```



Hide

```
ggsave(paste("./figure2/lmain_alpha", figi, ".pdf", sep = ''), device = "pdf")
```

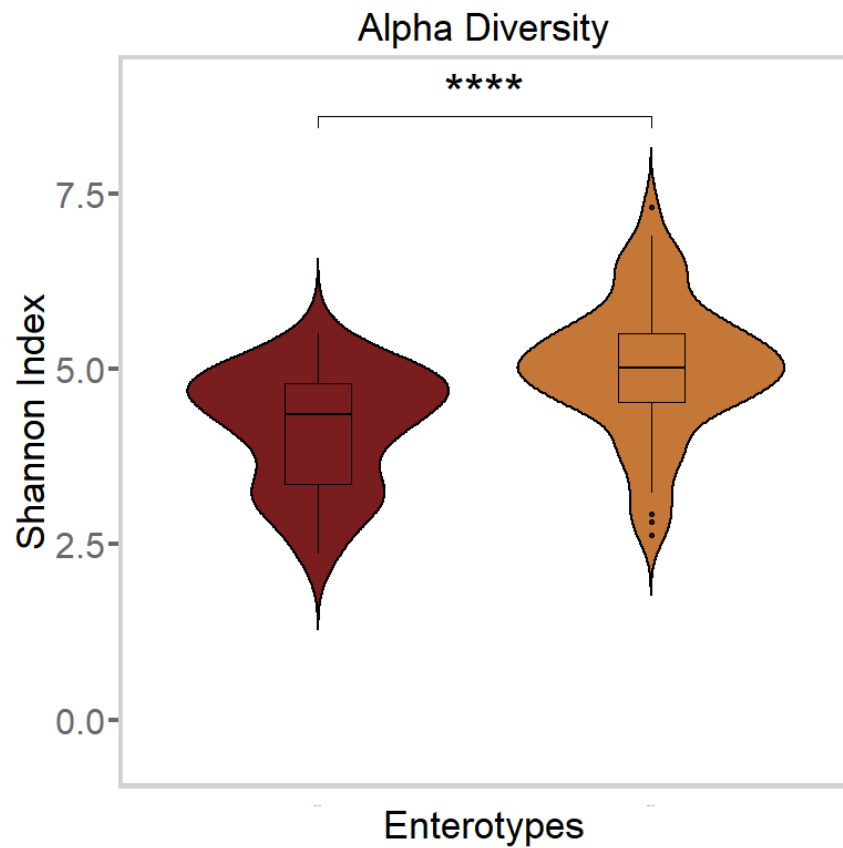
Saving 12.9 x 8 in image

Hide

```
figi = figi + 1
```

[Hide](#)

```
ggviolin(fmt_alpha[fmt_alpha$FMT %in% c('Before') & fmt_alpha$Disease %in% c('CD', 'UC')], x=
"Enterotypes", y="shannon", fill = "Enterotypes", add = "boxplot", alpha=1, add.params = list(a
lpha=0.8), palette = c("#7A1D1E", "#C47737"), size = 0.8, width = 0.8)+
  stat_compare_means(comparisons = list(c('before1', 'before2')), method = 'wilcox.test', label
= "p.signif", label.x = 1.5, label.y = 8.6, size=12)+
  scale_y_continuous(limits = c(-0.5, 9))+
  # yscale("log2", .format = FALSE)+
  theme_classic()+theme(legend.position = "right")+xlab(label = 'Enterotypes')+ylab("Shannon In
dex")+labs(title='Alpha Diversity')+
  theme(text=element_text(family = "sans", size=24), plot.title = element_text(size=26, hjust =
0.5), axis.text = element_text(size=24, color = 'dimgray'), axis.text.x = element_text(size=0),
axis.title.x = element_text(size=26), axis.title.y = element_text(size=26), axis.ticks.y = elem
ent_line(size=1.5, color = 'dimgray'), axis.ticks.length = unit(7, "pt"))+
  theme(aspect.ratio = 1, legend.background=element_blank(), legend.position=c(1.75, 0.6)
, panel.background = element_rect(fill = NA, colour = "lightgrey", size = 3)
, axis.line=element_line(colour=NA)
, legend.key = element_rect(fill = NA, color = NA), axis.ticks = element_blank())+
  guides(colour = guide_legend(override.aes = list(size=5)))
```

[Hide](#)

```
ggsave(paste("./figure2/1main_alpha", figi, ".pdf", sep = ''), device = "pdf")
```

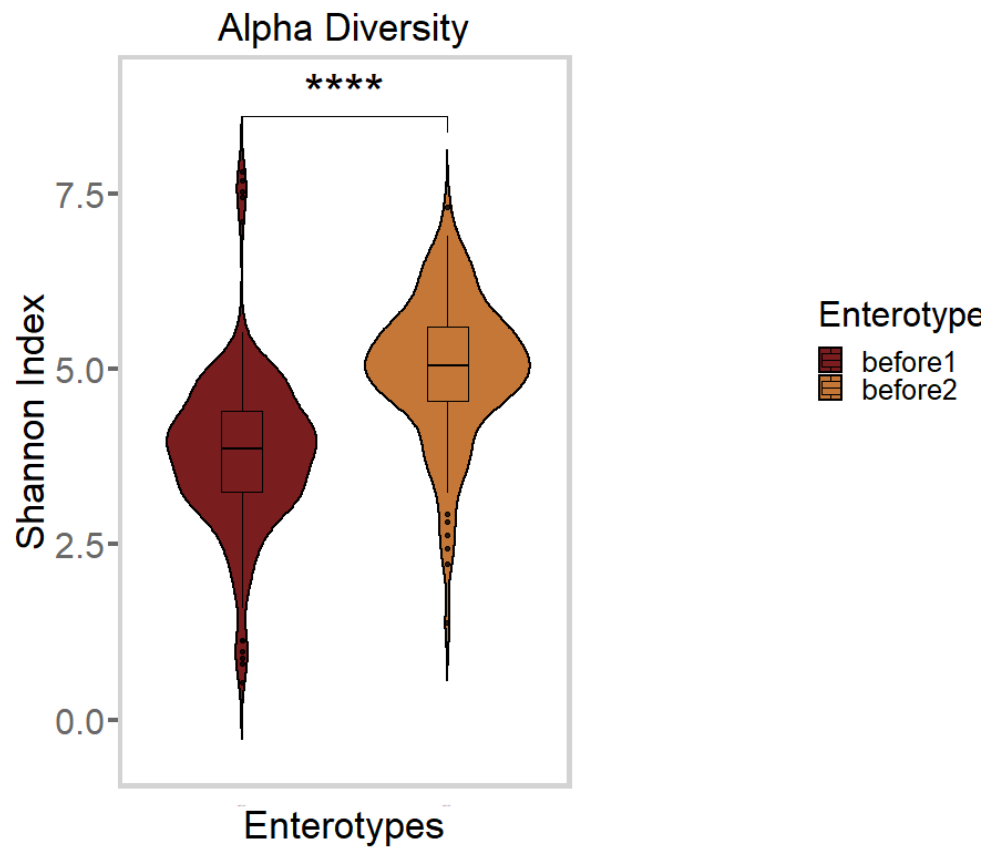
Saving 12.9 x 8 in image

[Hide](#)

```
figi = figi + 1
```

[Hide](#)

```
ggviolin(fmt_alpha[fmt_alpha$FMT %in% c('Before'),], x="Enterotypes", y="shannon", fill = "Enterotypes", add = "boxplot", alpha=1, add.params = list(alpha=0.8), palette = c("#7A1D1E", "#C47737"), size = 0.8, width = 0.8)+
  stat_compare_means(comparisons = list(c('before1', 'before2')), method = 'wilcox.test', label = "p.signif", label.x = 1.5, label.y = 8.6, size=12)+
  scale_y_continuous(limits = c(-0.5, 9))+
  # yscale("log2", .format = FALSE)+
  theme_classic()+theme(legend.position = "right")+xlab(label = 'Enterotypes')+ylab("Shannon Index")+labs(title='Alpha Diversity')+
  theme(text=element_text(family = "sans", size=24), plot.title = element_text(size=26, hjust = 0.5), axis.text = element_text(size=24, color = 'dimgray'), axis.text.x = element_text(size=0), axis.title.x = element_text(size=26), axis.title.y = element_text(size=26), axis.ticks.y = element_line(size=1.5, color = 'dimgray'), axis.ticks.length = unit(7, "pt"))+
  theme(aspect.ratio = 1.618, legend.background=element_blank(), legend.position=c(1.75, 0.6), panel.background = element_rect(fill = NA, colour = "lightgrey", size = 3), axis.line=element_line(colour=NA), legend.key = element_rect(fill = NA, color = NA), axis.ticks = element_blank())+
  guides(colour = guide_legend(override.aes = list(size=5)))
```

[Hide](#)

```
ggsave(paste("./figure2/main_", figi, ".pdf", sep = ''), device = "pdf")
```

Saving 12.9 x 8 in image

[Hide](#)

```
figi = figi + 1
```

[Hide](#)

```
##donor after distance
da_distance <- NULL
len <- nrow(meta_fil_config_entro)
L6_rela_fil_sAg_others_remove <- noise.removal(L6_rela_fil_sAg_others, percent=0.01)
```

```
[1] 0.01
```

[Hide](#)

```
for(i in 1:len){
  x1 <- as.numeric(as.character(L6_rela_fil_sAg_others_remove[, meta_fil_config_entro[i, "Don
or_sra"]]))
  x2 <- as.numeric(as.character(L6_rela_fil_sAg_others_remove[, meta_fil_config_entro[i, "SRA
_Sample"]]))
  da_distance <- rbind(da_distance, c(meta_fil_config_entro[i, "SRA_Sample"], vegdist(rbind(x
1, x2), method='bray'))))
}

colnames(da_distance) <- c('SRA_Sample', 'distance')

tmp_after <- meta_fil_config_entro[, c('SRA_Sample', 'Time', 'postfmt_symptoms', 'Dieasel')]

distance_after <- as.data.frame(merge(da_distance, tmp_after, by.x='SRA_Sample', by.y='SRA_Samp
le'), stringsAsFactors=F)
distance_after$distance <- as.numeric(as.character(distance_after$distance))
distance_after[is.na(distance_after[, 'Time']), 'Time']='NA'

distance_after$type <- 0

distance_after[distance_after$Time < 7, "type"] = 'Days'
distance_after[7 <= distance_after$Time & distance_after$Time < 31, "type"] = 'Weeks'
distance_after[31 <= distance_after$Time & distance_after$Time < 93, "type"] = 'Months'
distance_after[93 <= distance_after$Time, "type"] = 'Longterm'

distance_after$type <- as.factor(distance_after$type)

###
distance <- NULL
len <- nrow(meta_fil_config_entro)
L6_rela_fil_sAg_others_remove <- noise.removal(L6_rela_fil_sAg_others, percent=0.01)
```

```
[1] 0.01
```

[Hide](#)

```

for(i in 1:len){
  x1 <- as.numeric(as.character(L6_rela_fil_sAg_others_remove[, meta_fil_config_entro[i, "Don
or_sra"]]))
  x2 <- as.numeric(as.character(L6_rela_fil_sAg_others_remove[, meta_fil_config_entro[i, "Pre
vious_sra"]]))
  distance <- rbind(distance, c(meta_fil_config_entro[i, "SRA_Sample"], vegdist(rbind(x1, x
2), method='bray'))))
}

colnames(distance) <- c('SRA_Sample', 'distance')

# distance <- NULL
# len <- nrow(meta_fil_config_entro)
# L6_rela_fil_sAg_others_remove <- noise.removal(L6_rela_fil_sAg_others, percent=0.01)
# for(i in 1:len){
#   x1 <- as.numeric(as.character(L6_rela_fil_sAg_others_remove[, meta_fil_config_entro[i, "D
onor_sra"]]))
#   x2 <- as.numeric(as.character(L6_rela_fil_sAg_others_remove[, meta_fil_config_entro[i, "P
revious_sra"]]))
#   distance <- rbind(da_distance, c(meta_fil_config_entro[i, "SRA_Sample"], vegdist(rbind(x
1, x2), method='bray'))))
# }

distance_before <- as.data.frame(distance)
distance_before$Time <- 'before'
distance_before$postfmt_symptoms <- 'before'
distance_before$type <- 'Before'
distance_before$Dieasel <- meta_fil_config_entro[, "Dieasel"]

rbind(distance_after, distance_before) -> distance_after_before

distance_after_before$type = factor(distance_after_before$type, levels=c('Before', 'Days', 'Wee
ks', 'Months', 'Longterm'))
distance_after_before$distance <- as.numeric(as.character(distance_after_before$distance))

set.seed(100)

```

[Hide](#)

```

distance_after_before_et1 <- distance_after_before[distance_after_before$SRA_Sample %in% c(as.c
haracter(meta_fil_config_entro[meta_fil_config_entro$pre_entro %in% c('before1'), 'SRA_Sample'
]),],]
distance_after_before_et2 <- distance_after_before[distance_after_before$SRA_Sample %in% c(as.c
haracter(meta_fil_config_entro[meta_fil_config_entro$pre_entro %in% c('before2'), 'SRA_Sample'
]),],]

dim(distance_after_before_et1)

```

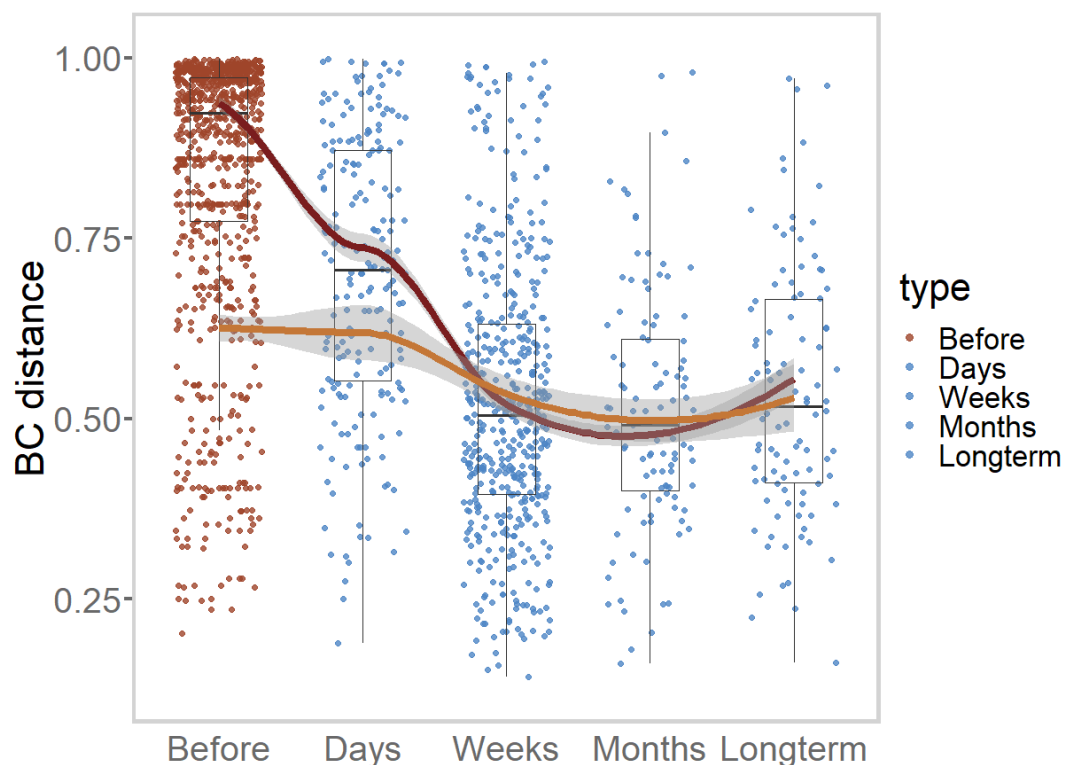
```
[1] 1202    6
```

[Hide](#)


```

ggplot()+
  geom_jitter(aes(x = type, y = distance, col=type), distance_after_before, size=2, alpha=0.8,
width = 0.3)+
  geom_boxplot(aes(x = type, y = distance), distance_after_before, alpha=0, width=0.4, size=0.6
)+
  geom_smooth(aes(x=as.numeric(type), y=distance), level = 0.9, distance_after_before_et1, si
ze=3, color='#7A1D1E', method='loess')+
  geom_smooth(aes(x=as.numeric(type), y=distance), level = 0.9, distance_after_before_et2, si
ze=3, color='#C47737', method='loess')+
  labs(x= c(''), y=c('BC distance'), title = c(''))+#Distance between donor and recipient
  scale_colour_manual(values=c("#9F452A", "#4E86C6", "#4E86C6", "#4E86C6", "#4E86C6"))+#962E2B
  scale_y_continuous(expand=c(0.04, 0.03))+
  # theme(text=element_text(family = "sans", size=24), plot.title = element_text(size=24, hjust
t = 0.5), axis.text.x = element_text(size=21, color = 'dimgray'))+
  theme(text=element_text(family = "sans", size=32), plot.title = element_text(size=34, hjust =
0.5), axis.text = element_text(size=29, color = 'dimgray'), axis.title.x = element_text(size=34
), axis.title.y = element_text(size=34), axis.ticks.y = element_line(size=1.5, color = 'dimgray'
), axis.ticks.length = unit(7, "pt"))+
  theme(aspect.ratio = 0.95, legend.background=element_blank(), legend.position=c(1.75, 0.6)
, panel.background = element_rect(fill = NA, colour = "lightgrey", size = 3)
, axis.line=element_line(colour=NA), axis.ticks.x = element_blank()
, legend.key = element_rect(fill = NA, color = NA))+
  # theme(axis.text.y = element_blank(), axis.ticks.y = element_blank(), axis.title.y = element
_blank(),)+
  guides(colour = guide_legend(override.aes = list(size=3)))+theme(panel.grid.major=element_bla
nk(), panel.grid.minor=element_blank())

```



Hide

```
ggsave(paste("../figure2/2main_distance", figi, ".pdf", sep = ''), device = "pdf", useDingbats=FALSE)
```

Saving 16.2 x 10 in image

Hide

```
figi = figi + 1
```

Hide

```
###After FMT
```

Hide

```
all_distance <- NULL

# meta_fil_config_end_na <- meta_fil_config_entro[!meta_fil_config_entro$postfmt_symptoms %in%
  c(NA),]
meta_fil_config_end_na <- meta_fil_config1_na#meta_fil_config_entro[!meta_fil_config_entro$post
fmt_symptoms %in% c(NA),]
L6_rela_fil_sAg_others_remove <- noise.removal(L6_rela_fil_sAg_others, percent=0.01)
```

```
[1] 0.01
```

Hide

```

len <- nrow(meta_fil_config_end_na)
for(i in 1:len){
  x1 <- as.numeric(as.character(L6_rela_fil_sAg_others_remove[, meta_fil_config_end_na[i, "Donor_sra"]]))
  x2 <- as.numeric(as.character(L6_rela_fil_sAg_others_remove[, meta_fil_config_end_na[i, "SRA_Sample"]]))
  x3 <- as.numeric(as.character(L6_rela_fil_sAg_others_remove[, meta_fil_config_end_na[i, "Previous_sra"]]))
  all_distance <- rbind(all_distance, c(vegdist(rbind(x1, x2), method='bray'), vegdist(rbind(x1, x3), method='bray'), (vegdist(rbind(x1, x3), method='bray') - vegdist(rbind(x1, x2), method='bray')), vegdist(rbind(x2, x3), method='bray'))))
}

# all_oth_health_distance <- NULL
# for(i in 1:len){
#   x2 <- as.numeric(as.character(L6_rela_fil_sAg_others_remove[, meta_fil_config_end_na[i, "SRA_Sample"]]))
#   x3 <- as.numeric(as.character(L6_rela_fil_sAg_others_remove[, meta_fil_config_end_na[i, "Previous_sra"]]))
#   post <- meta_fil_config_end_na[i, "postfmt_symptoms"]
#   for(j in 1:len){
#     if(i == j){next}
#     x1 <- as.numeric(as.character(L6_rela_fil_sAg_others_remove[, meta_fil_config_end_na[j, "Donor_sra"]]))
#
#     all_oth_health_distance <- rbind(all_oth_health_distance, c(vegdist(rbind(x1, x2), method='bray'), vegdist(rbind(x1, x3), method='bray'), (vegdist(rbind(x1, x3), method='bray') - vegdist(rbind(x1, x2), method='bray')), post))
#   }
# }
dim(meta_fil_config_end_na)

```

```
[1] 286 19
```

[Hide](#)

```
# dim(all_oth_health_distance)
```

[Hide](#)

```

colnames(all_distance) <- c('da_d', 'dp_d', 'diff_distance', 'pa_d')
# auc_ci <- roc(meta_fil_config_end_na$postfmt_symptoms, all_distance[, 'diff_distance'])
# auc_ci
test_coin <- as.data.frame(cbind(all_distance[, ], meta_fil_config_end_na$SRA_Sample, meta_fil_config_end_na$postfmt_symptoms, meta_fil_config_end_na$PRJ, meta_fil_config_end_na$Disease))
colnames(test_coin) <- c('da_d', 'dp_d', 'dista', 'pa_d', 'SRA_Sample', 'post', 'PRJ', 'Disease')
test_coin$post <- ifelse(test_coin$post == 'failure', 'Failure', 'Response')
test_coin$post <- factor(test_coin$post, levels = c('Response', 'Failure'))
test_coin$dista <- as.numeric(as.character(test_coin$dista))
test_coin$da_d <- as.numeric(as.character(test_coin$da_d))
test_coin$dp_d <- as.numeric(as.character(test_coin$dp_d))
test_coin$pa_d <- as.numeric(as.character(test_coin$pa_d))

```

[Hide](#)

```
library(ggpubr)
library(ggbeeswarm)
```

```
test_coin_post <- test_coin %>% group_by(post) %>% dplyr::summarise(mean_dis = (mean(da_d)), sd
_dis = 1.96*sd(da_d)/sqrt(length(da_d)))
```

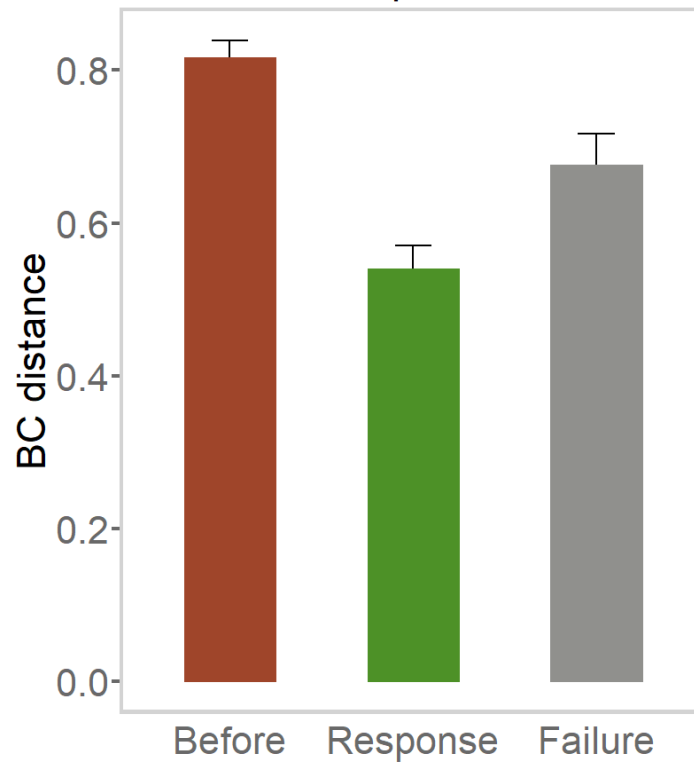
```
`summarise()` ungrouping output (override with `.groups` argument)
```

[Hide](#)

```
test_coin_post_a <- test_coin_post %>% add_row(mean_dis=mean(test_coin$dp_d), sd_dis=1.96*sd(te
st_coin$dp_d)/sqrt(length(test_coin$dp_d)))
test_coin_post_a$post <- c('Response', 'Failure', 'Before')
test_coin_post_a$post <- factor(test_coin_post_a$post, levels = c('Before', 'Response', 'Failur
e'))

ggbarplot(test_coin_post_a, x='post', y='mean_dis', color = 'NA', stat="identity", width = 0)+
  geom_errorbar(aes(ymin=mean_dis-sd_dis, ymax=mean_dis+sd_dis), width=.2, position=position_do
dge(.1), size=1)+
  geom_bar(aes(x=post, y=mean_dis, fill = post, color=post), stat="identity", width = 0.5, alph
a=1)+
  # geom_jitter(aes(post, da_d, color=post), test_coin, size=3, width = 0.15)+
  # geom_quasirandom(aes(post, da_d, color=post), test_coin, size=3, method='smiley')+
  # geom_boxplot(aes(post, da_d), test_coin, alpha=0, width=0.3)+
  labs(x= c(''), y=c('BC distance'), title = c('Donor to recipient after FMT'))+
  scale_fill_manual(values=c("#9F452A", "#4D9127", "#90908D"))+#
  scale_colour_manual(values=c("#9F452A", "#4D9127", "#90908D"))+#"#962E2B", "#4E86C6", "#4E86C
6", "#4E86C6", "#4E86C6"))+
  # theme(text=element_text(family = "sans", size=24), plot.title = element_text(size=24, hjust
= 0.5), axis.text.x = element_text(size=21, color = 'dimgray'))+
  theme(text=element_text(family = "sans", size=32), plot.title = element_text(size=34, hjust =
0.5), axis.text = element_text(size=32, color = 'dimgray'), axis.title.x = element_text(size=34
), axis.title.y = element_text(size=34), axis.ticks.y = element_line(size=1.5, color = 'dimgray'
), axis.ticks.length = unit(7, "pt"))+
  theme(aspect.ratio = 1.2, legend.background=element_blank(), legend.position=c(1.75, 0.6)
, panel.background = element_rect(fill = NA, colour = "lightgrey", size = 3)
, axis.line=element_line(colour="lightgrey"), axis.ticks.x = element_blank()
, legend.key = element_rect(fill = NA, color = NA))+
  # theme(axis.text.y = element_blank(), axis.ticks.y = element_blank(), axis.title.y = element
_blank(),)+
  guides(colour = guide_legend(override.aes = list(size=3)))
```

Donor to recipient after FMT



Hide

```
test_coin_et1<-test_coin[test_coin$SRA_Sample %in% c(as.character(meta_fil_config_entro[meta_fil_config_entro$pre_entro %in% c('before1'), 'SRA_Sample'])), ]
test_coin_et2<-test_coin[test_coin$SRA_Sample %in% c(as.character(meta_fil_config_entro[meta_fil_config_entro$pre_entro %in% c('before2'), 'SRA_Sample'])), ]

ggsave(paste("./figure2/fig2_ss", figi, ".pdf", sep = ''), device = "pdf", useDingbats=FALSE)
```

Saving 16.2 x 10 in image

Hide

```
figi = figi + 1
```

Hide

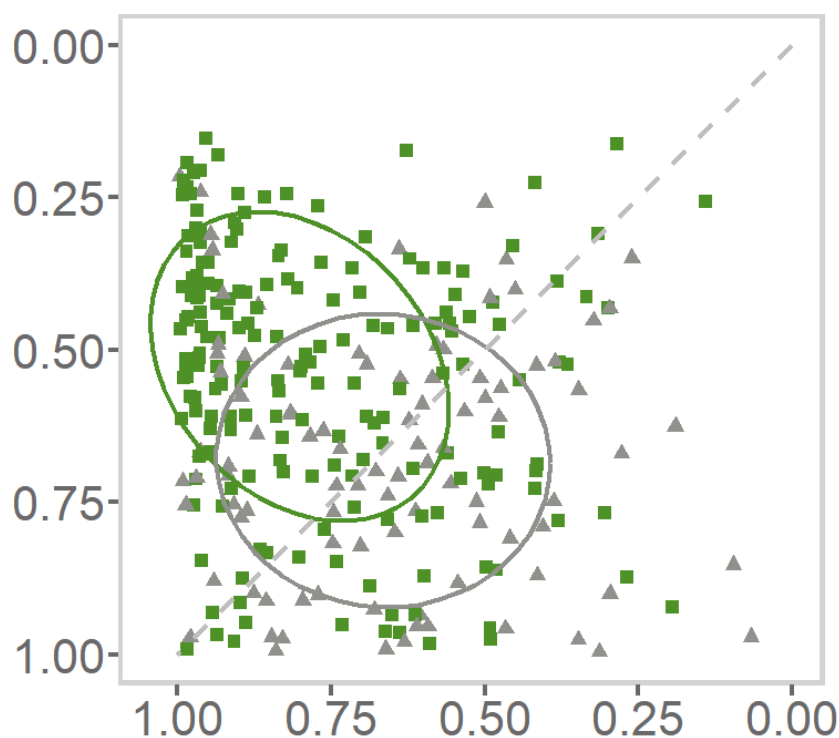
```
#plot triangle
#test_coin
```

Hide

```

ggplot(test_coin, aes(pa_d, da_d))+
  geom_point(aes(color=post, fill=post, shape=post), size=4, alpha=1)+
  theme_classic()+
  scale_x_reverse()+
  scale_y_reverse()+
  stat_ellipse(aes(colour=as.factor(post), group=as.factor(post)), level=0.6, size=1.6, alpha=1
, show.legend = NA)+
  geom_segment(aes(x=0, xend=1, y=0, yend=1), size=1.5, color='grey', linetype='dashed')+
  labs(title="", x="", y="")+
  theme(text=element_text(family = "sans", size=32), plot.title = element_text(size=34, hjust =
0.5), axis.text = element_text(size=32, color = 'dimgray'), axis.title.x = element_text(size=34
), axis.title.y = element_text(size=34), axis.ticks = element_line(size=1.5, color = 'dimgray'),
axis.ticks.length = unit(7, "pt"))+
  theme(aspect.ratio=0.95, legend.position = c(4, .65), legend.background=element_rect(fill = N
A), legend.text = element_text(size=18))+
  scale_color_manual(breaks=c('Response_p', 'Failure_p', 'Response', 'Failure'), values=c("#4D9
127", "#90908D", "#4D9127", '#90908D'))+
  scale_fill_manual(breaks=c('Response_p', 'Failure_p', 'Response', 'Failure'), values=c("#4D91
27", "#90908D", "#4D9127", '#90908D'))+ #, "#90908D", "#4D9127", "#7A1D1E", "#C47737", "#E7A6
00"
  scale_shape_manual(values=c(22, 17))+
  theme(panel.background = element_rect(fill = NA, colour = "lightgrey", size = 3) #lightgrey
, axis.line=element_line(colour="lightgrey"))+theme(panel.grid = element_blank())

```



Hide

```

library(vegan)
adonis(test_coin[,c('pa_d', 'da_d')] ~ post, test_coin, permutations = 999)

```

Call:

```
adonis(formula = test_coin[, c("pa_d", "da_d")] ~ post, data = test_coin, permutations = 999)
```

Permutation: free

Number of permutations: 999

Terms added sequentially (first to last)

	Df	SumsOfSqs	MeanSqs	F.Model	R2	Pr(>F)
post	1	0.5809	0.58088	24.428	0.0792	0.001 ***
Residuals	284	6.7533	0.02378		0.9208	
Total	285	7.3342			1.0000	

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Hide

```
anosim(vegdist(test_coin[,c('pa_d', 'da_d')]), test_coin[, 'post'], test_coin, permutations = 999)
```

Call:

```
anosim(x = vegdist(test_coin[, c("pa_d", "da_d")]), grouping = test_coin[, "post"], permutations = 999, distance = test_coin)
```

Dissimilarity: bray

ANOSIM statistic R: 0.1011

Significance: 0.001

Permutation: free

Number of permutations: 999

Hide

```
ggsave(filename = 'figure2/2main_triangle.pdf', device = "pdf", width = 8, useDingbats=FALSE)
```

Saving 8 x 8 in image

Hide

```
adonis(test_coin[,c('pa_d', 'da_d')] ~ post, test_coin, permutations = 999)
```

Call:

```
adonis(formula = test_coin[, c("pa_d", "da_d")] ~ post, data = test_coin, permutations = 999)
```

Permutation: free

Number of permutations: 999

Terms added sequentially (first to last)

	Df	SumsOfSqs	MeanSqs	F.Model	R2	Pr(>F)
post	1	0.5809	0.58088	24.428	0.0792	0.001 ***
Residuals	284	6.7533	0.02378		0.9208	
Total	285	7.3342			1.0000	

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Hide

```
anosim(vegdist(test_coin[, c('pa_d', 'da_d')], method="manhattan"), test_coin[, 'post'], test_coin, permutations = 999)
```

Call:

```
anosim(x = vegdist(test_coin[, c("pa_d", "da_d")], method = "manhattan"), grouping = test_coin[, "post"], permutations = 999, distance = test_coin)
```

Dissimilarity: manhattan

ANOSIM statistic R: 0.1107

Significance: 0.001

Permutation: free

Number of permutations: 999

Hide

```
donor_before_after_color <- c("#9F452A", "#4E86C6", "#235E27")
```

Hide


```

##before after validate close to donor through genus
#genus: L6_rela_fil_sAg_remove L6_rela_fil_sAg_remove_simp
#config: meta_fil_config1
#

library(reshape2)
library(ggplot2)

coord_radar <- function(theta="x", start=0, direction = 1){
  theta<- match.arg(theta, c('x', 'y'))
  r<-if(theta=='x'){
    'y'
  }else{ 'x' }
  ggproto("CoordRadar", CoordPolar, theta = theta, r = r, start = start, direction = sign(direction), is_linear = function(coord) TRUE)
}

select_plot_genus <- function(search_genus, pre_entro){
  searched_abundance <- L6_rela_fil_sAg_remove_simp[search_genus,]

  ###build two column selected metafile
  tmp_3column_before1 <- meta_fil_config1[meta_fil_config1$pre_entro %in% c(pre_entro), c('Previous_sra', 'SRA_Sample', 'Donor_sra')]
  colnames(tmp_3column_before1) <- c('Before', 'After', 'Donor')
  FMTstage_before1 <- melt(tmp_3column_before1, measure.vars = c('Before', 'After', 'Donor'))
  FMTstage_before1$variable <- factor(FMTstage_before1$variable, levels = c('Before', 'After', 'Donor'))

  ###select
  FMTstage_abun_before1 <- cbind(FMTstage_before1, searched_abundance[FMTstage_before1$value])
  colnames(FMTstage_abun_before1) <- c(colnames(FMTstage_before1), 'abun')

  # ggplot()+
  # geom_point(aes(x = variable, y = log(abun + 1), color = variable), position = position_jitter(w = 0.35, h = 0.1), size=2, FMTstage_abun_before1)+
  # geom_boxplot(aes(x = variable, y = log(abun + 1)), FMTstage_abun_before1, color='black', alpha=0, size=0.8)+
  # labs(x= c(''), y=c('Log Abundance'), title = c(search_genus))+
  # scale_colour_manual(name="FMT", values=c(donor_before_after_color, "#962E2B", "#4E86C6", "#4D9127", "#90908D", 'lightgrey'))+ '#C77CFF', '#43AFC8',
  # theme(text=element_text(family="sans", size=28), plot.title = element_text(size=24, hjust = 0.5), axis.title.x = element_text(size=24, vjust = -0.5, hjust = 0.71, color = 'dimgray'))+
  # theme(aspect.ratio = 0.95, legend.background=element_blank(), legend.position=c(1.75, 0.6)
  # ,panel.background = element_rect(fill = NA, colour = "lightgrey", size = 3)
  # ,axis.line=element_line(colour="lightgrey")
  # ,legend.key = element_rect(fill = NA, color = NA))+
  # guides(colour = guide_legend(override.aes = list(size=5)))
  stat_df <- FMTstage_abun_before1 %>% dplyr::group_by(variable) %>% dplyr::summarise(mean_abun = (mean(abun)), sd_abun = 1.96*sd(abun)/sqrt(length(abun)))

  print(stat_df[, c('variable', 'mean_abun')])
  ggplot(stat_df, aes(x = variable, y = mean_abun))+
    geom_errorbar(aes(ymin=mean_abun-sd_abun, ymax=mean_abun+sd_abun), width=.2, position=position_dodge(.1), size=1)+
    geom_bar(aes(fill = variable), stat="identity", width = 0.6, alpha=1)+ #position = position_jitter(w = 0.35, h = 0.1), size=2,

```

```

# geom_boxplot(aes(x = variable, y = log(abun + 1)), FMTstage_abun_before1, color='black',
alpha=0, size=0.8)+
  labs(x= c(''), y=c('Abundance'), title = c(search_genus))+
  #scale_colour_manual(name="FMT", values=c("#962E2B", "#4E86C6", "#4D9127", "#90908D", 'lightgrey'))+
  # "#C77CFF", '#43AFC8',
  scale_fill_manual(name="FMT", values=c(donor_before_after_color, "#90908D", 'lightgrey'))+
  scale_y_continuous(expand = expansion(mult=c(0.03, 0.15)))+
  # theme(text=element_text(family = "sans", size=28), plot.title = element_text(size=32, hjust = 0.5, face = "italic"), axis.title.x = element_text(size=24, vjust = -0.5, hjust = 0.71, color = 'dimgray'), axis.title = element_text(size=31))+
  theme(text=element_text(family = "sans", size=32), plot.title = element_text(size=34, hjust = 0.5, face = "italic"), axis.text = element_text(size=32, color = 'dimgray'), axis.title.x = element_text(size=34), axis.title.y = element_text(size=34), axis.ticks = element_blank()+
  theme(aspect.ratio = 0.618, legend.background=element_blank()#, legend.position=c(1.75, 0.6)
    ,panel.background = element_rect(fill = NA, colour = "lightgrey", size = 3)
    ,axis.line=element_line(colour="lightgrey")
    ,legend.key = element_rect(fill = NA, color = NA))+
  guides(colour = guide_legend(override.aes = list(size=5)))

}

```

[Hide](#)

```

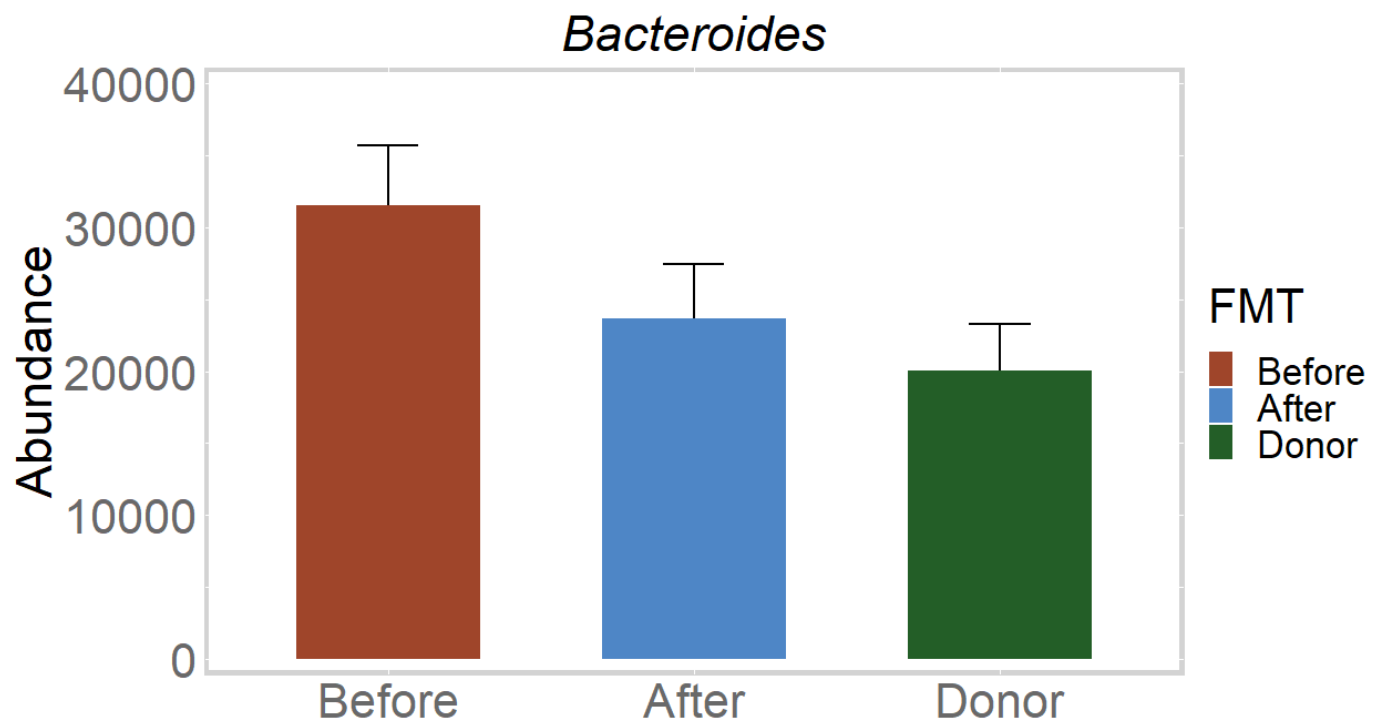
# pdf('supple_no/fig2_genus_change.pdf', height = 8, width=8)

select_plot_genus("Bacteroides", "before2")

```

```
`summarise()` ungrouping output (override with `.groups` argument)
```

variable <fctr>	mean_abun <dbl>
Before	31502.44
After	23652.37
Donor	20050.70
3 rows	



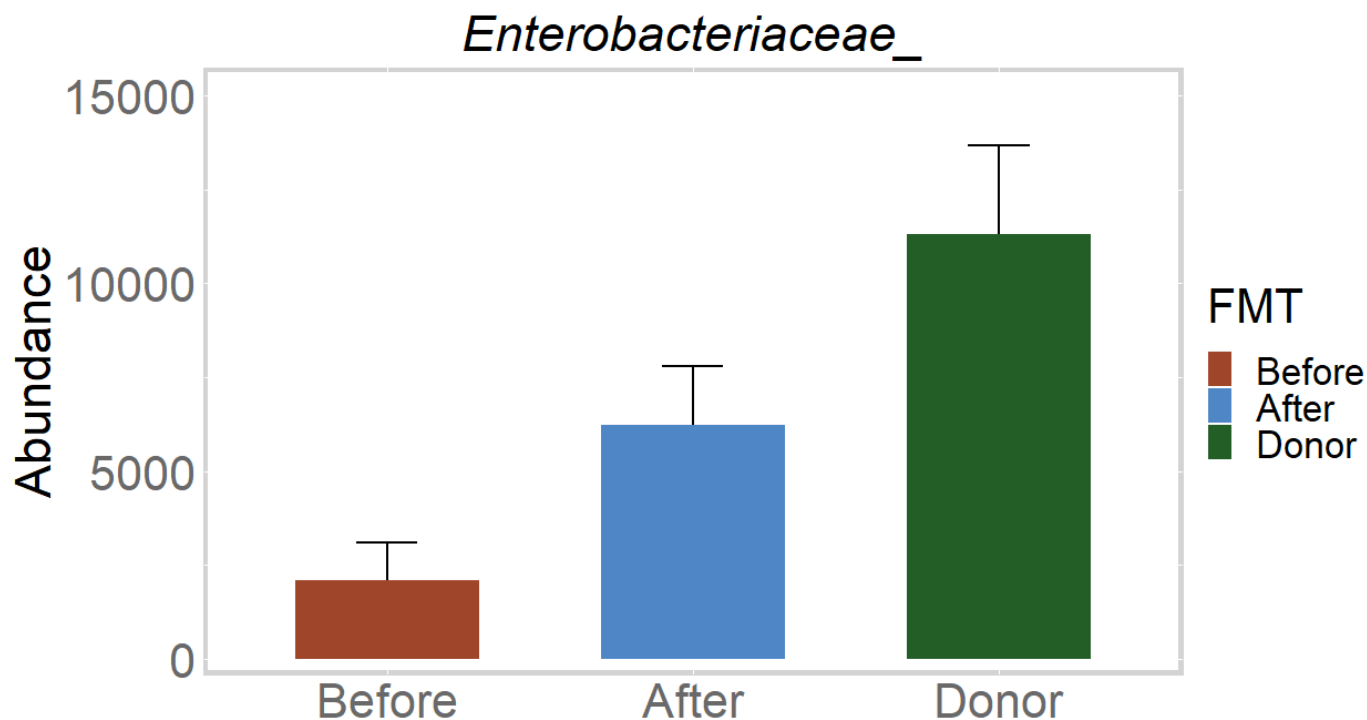
Hide

```
select_plot_genus("Enterobacteriaceae_", "before1")
```

```
`summarise()` ungrouping output (override with `.groups` argument)
```

variable <fctr>	mean_abun <dbl>
Before	2093.719
After	6215.397
Donor	11301.136

3 rows



Hide

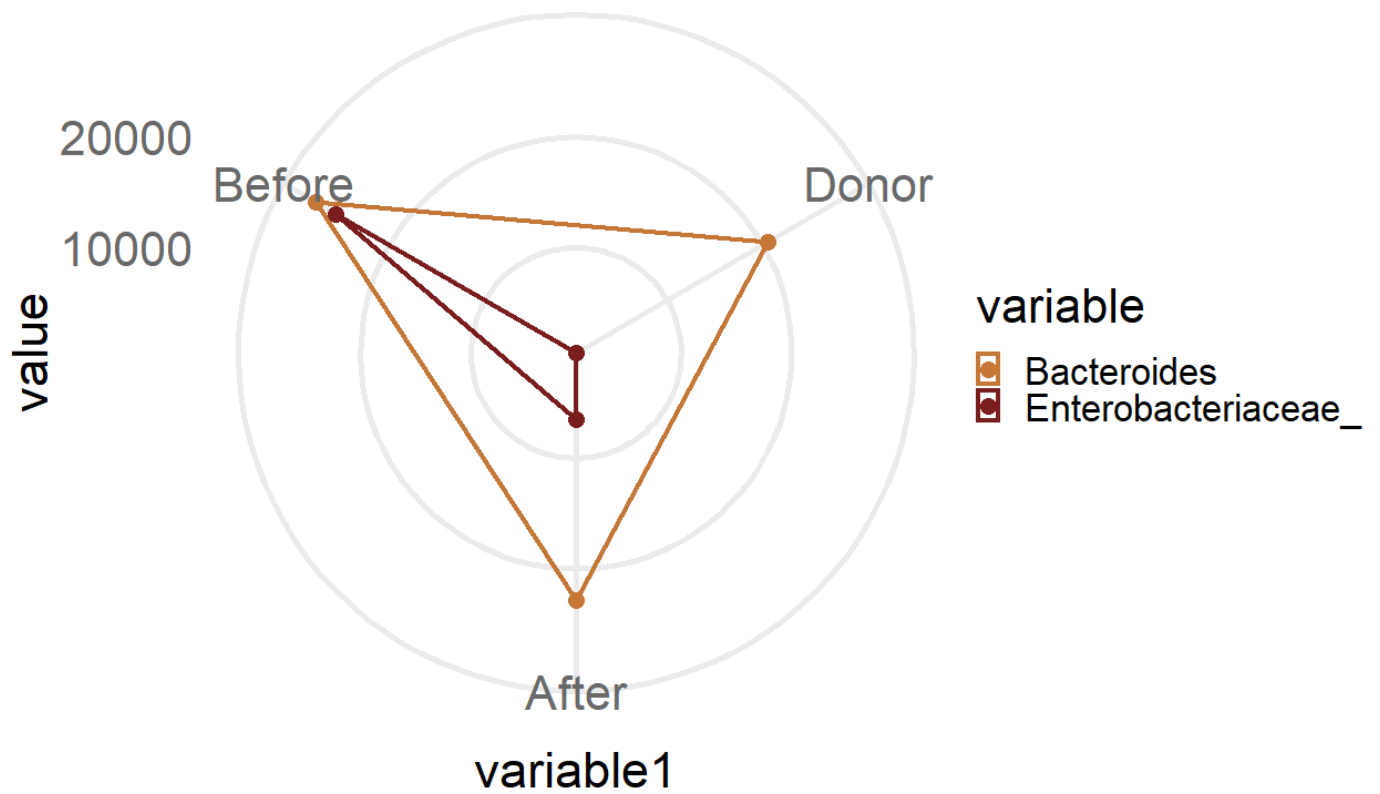
```
# dev.off()
```

Hide

```
stat_df_rader <- data.frame(variable1=c('Before', 'After', 'Donor'), Bacteroides=c(27682.54, 22915.07, 20521.38), Enterobacteriaceae_=c(25639.8970, 6512.8678, 474.3158), stringsAsFactors = F)
```

```
library(reshape2)
stat_df_rader_m <- melt(stat_df_rader, measure.vars = c('Bacteroides', 'Enterobacteriaceae_'))
stat_df_rader_m$variable1 <- factor(stat_df_rader_m$variable1, levels=c('Donor', 'After', 'Before'))
```

```
ggplot(data=stat_df_rader_m, aes(x=variable1, y=value, group=variable))+
  geom_polygon(aes(color=variable), fill='NA', size=1.5)+
  geom_point(aes(fill=variable), size=5, shape=21, color='NA')+
  coord_radar()+
  # ylim()+
  theme_minimal()+
  scale_color_manual(values=c("#C47737", "#7A1D1E"))+
  scale_fill_manual(values=c("#C47737", "#7A1D1E", donor_before_after_color))+
  theme(text=element_text(family = "sans", size=32), plot.title = element_text(size=34, hjust = 0.5, face = "italic"), axis.text = element_text(size=32, color = 'dimgray'), axis.title.x = element_text(size=34), axis.title.y = element_text(size=34), axis.ticks = element_blank(), panel.grid = element_line(size=2))
```



Hide

```
ggsave(filename = './figure2/2main_sig_rader.pdf', device = "pdf", useDingbats=FALSE)
```

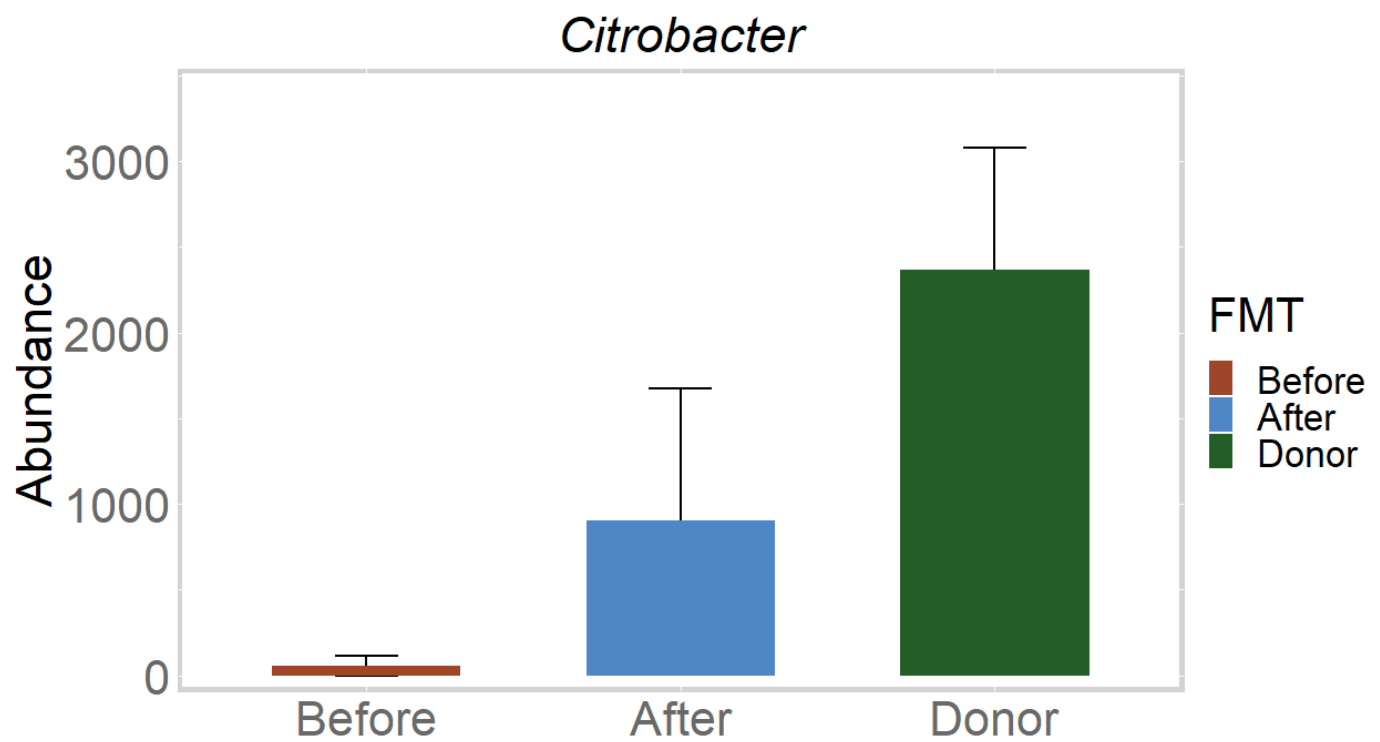
Saving 12.9 x 8 in image

Hide

```
# pdf('./figure2/fig_s4_genus_change.pdf', height = 8, width=8)
select_plot_genus("Citrobacter", "before1")
```

```
`summarise()` ungrouping output (override with `.groups` argument)
```

variable <fctr>	mean_abun <dbl>
Before	59.17878
After	901.00154
Donor	2368.72863
3 rows	



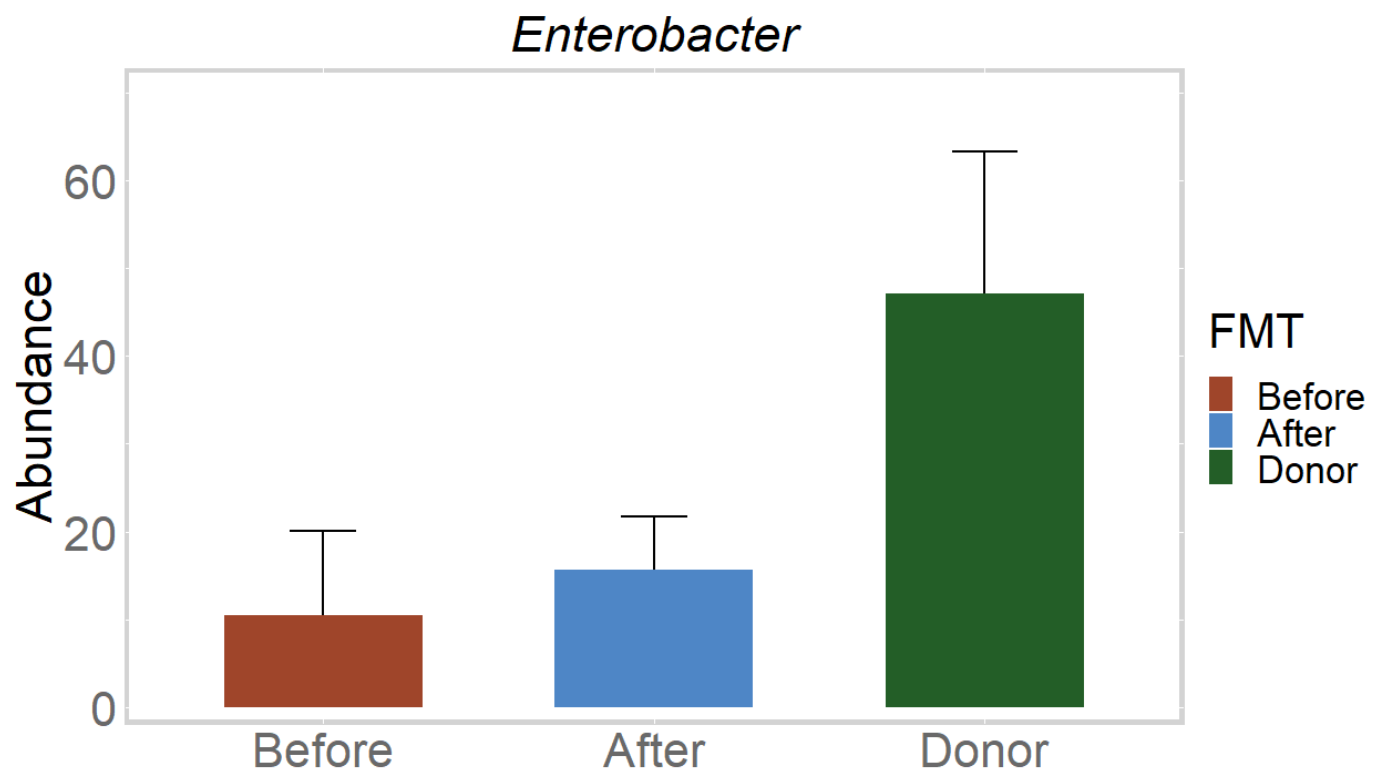
Hide

```
select_plot_genus("Enterobacter", "before1")
```

```
`summarise()` ungrouping output (override with `.groups` argument)
```

variable <fctr>	mean_abun <dbl>
Before	10.48838
After	15.62222
Donor	47.11067

3 rows



Hide

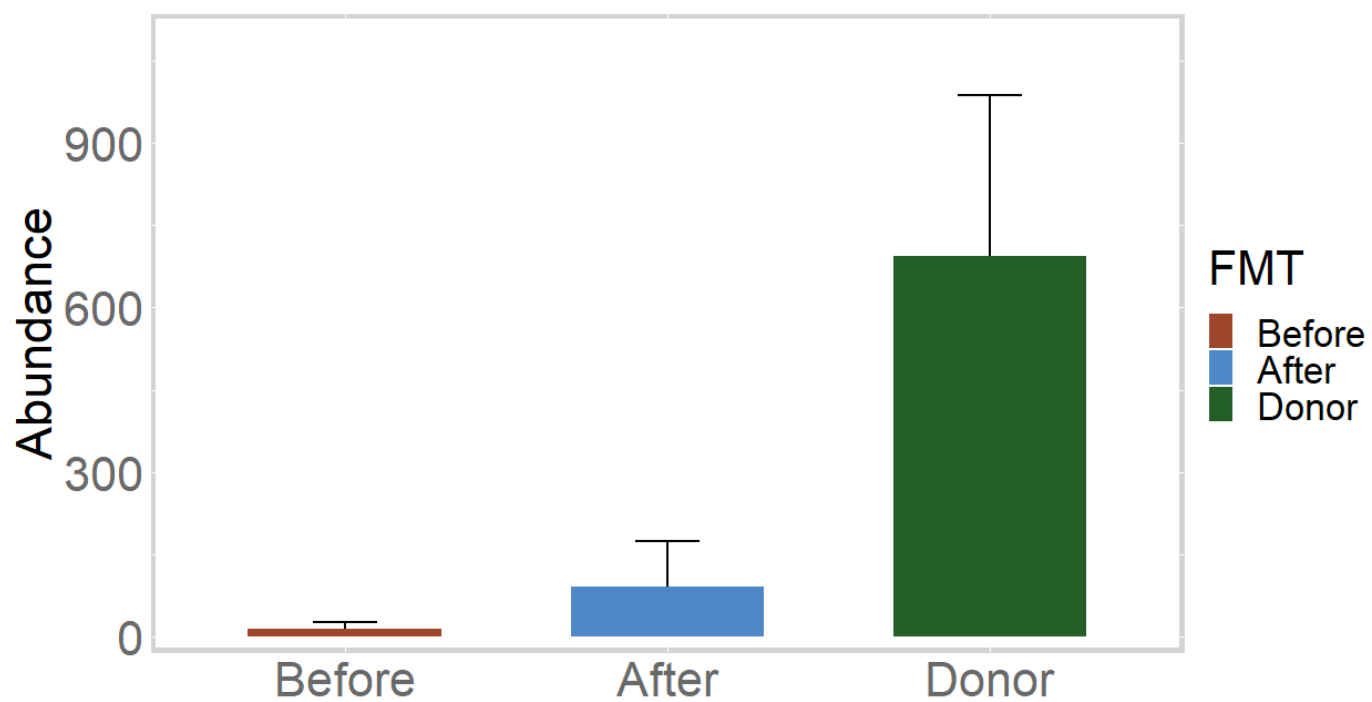
```
select_plot_genus("Acinetobacter", "before1")
```

```
`summarise()` ungrouping output (override with `.groups` argument)
```

variable <fctr>	mean_abun <dbl>
Before	13.93075
After	90.80443
Donor	693.50879

3 rows

Acinetobacter



Hide

dev.off()

Hide


```

library(ggbeeswarm)
library(ggpubr)
###marked genus in response come from donor
#genus: L6_rela_fil_sAg_remove L6_rela_fil_sAg_remove_simp
#config: meta_fil_config1_na
#
library(coin)
library(reshape2)
response_abundance <- function(pre_entro){
  ###marked genus in after response group
  # pre_entro <- c('before1', 'before2')
  tmp_after_response <- meta_fil_config1_na[meta_fil_config1_na$pre_entro %in% c(pre_entro),
c('SRA_Sample', 'postfmt_symptoms', 'PRJ')]

  L6_rela_fil_sAg_remove_simp_after <- L6_rela_fil_sAg_remove_simp[,tmp_after_response$SRA_Sa
mple]

  # cols <- ncol(feature_abun_dat)
  group <- tmp_after_response$postfmt_symptoms
  prj <- tmp_after_response$PRJ
  seq(0.1, 0.9, 0.05) -> quan

  tmp_pval <- apply(L6_rela_fil_sAg_remove_simp_after, 1, function(x){
    pt <- as.data.frame(cbind(x, group, prj))
    colnames(pt)<-c('nx', 'group', 'prj')
    # upt <- unique(pt)
    upt <- pt
    prj_list <- upt$prj
    list <- NULL
    for(i in unique(prj_list)){
      if (length(prj_list[prj_list %in% i]) > 2){
        list <- c(list, i)
      }
    }
    upt <- upt[upt$prj %in% list,]
    upt$nx <- as.numeric(as.character(upt$nx))
    tmp_test <- wilcox_test(nx ~ group | prj, upt)
    pval_a <- 1
    pval_a <- pvalue(tmp_test)
    if(is.na(pval_a)){pval_a<-1}
    a_nx <- upt[upt$group %in% c('response'), 'nx']
    p_nx <- upt[upt$group %in% c('failure'), 'nx']
    case <- quantile(log10(a_nx + 0.0001), quan)
    control <- quantile(log10(p_nx+ 0.0001), quan)
    gfc <- sum((case - control))/length(quan)
    return(c(pval_a, gfc))
  })

  ###select marked genus qvalue<0.05, and combine abundance
  tmp_pval_t <- data.frame(t(tmp_pval))
  colnames(tmp_pval_t) <- c('pval', 'gfc')
  # tmp_pval_t$id <- rownames(tmp_pval_t)
  # entero_diff_t$pval <- as.numeric(as.character(entero_diff_t$pval))
  tmp_qvalue <- p.adjust(tmp_pval_t$pval, method='fdr')
  tmp_pval_adjust <- cbind(tmp_pval_t, tmp_qvalue)
  tmp_pval_adjust05 <- tmp_pval_adjust[(tmp_pval_adjust$tmp_qvalue < 0.05),]#tmp_qvalue < 0.0
5,]

```

```

tmp_L6_05 <- L6_rela_fil_sAg_remove_simp[rownames(tmp_pval_adjust05),]
tmp_L6_after_05 <- L6_rela_fil_sAg_remove_simp_after[rownames(tmp_pval_adjust05),]

#tmp_L6_after_05  tmp_pval_adjust05
#tmp_after_response
mean_after_response <- apply(tmp_L6_after_05[,tmp_after_response$postfmt_symptoms %in% c('response')], 1, function(x){(mean(x))})#quantile((x + 0.0001), quan)
mean_after_failure <- apply(tmp_L6_after_05[,tmp_after_response$postfmt_symptoms %in% c('failure')], 1, function(x){(mean(x))})
plot_after_response <- as.data.frame(cbind(rownames(tmp_L6_after_05), as.numeric((mean_after_response)), as.numeric((mean_after_failure))), stringsAsFactors = F)
colnames(plot_after_response) <- c('genus', 'response', 'failure')
plot_after_response$response <- (as.numeric(plot_after_response$response)+0)
plot_after_response$failure <- (as.numeric(plot_after_response$failure)+0)
plot_after_response$genus <- factor(plot_after_response$genus, levels = (rownames(tmp_pval_adjust05)[order(sign(tmp_pval_adjust05$gfc)/tmp_pval_adjust05$tmp_qvalue, decreasing = F)]))
plot_after_response$diff <- log2(plot_after_response$failure / plot_after_response$response)#ifelse(plot_after_response$response > plot_after_response$failure, plot_after_response$response / plot_after_response$failure, plot_after_response$failure / plot_after_response$response)
plot_after_response$qvalue <- tmp_pval_adjust05$tmp_qvalue
min_y = 0
mmin_y = -2

library(reshape2)
plot_after_response <- plot_after_response[order(ifelse(plot_after_response$failure < plot_after_response$response, plot_after_response$response, -1*plot_after_response$failure)),]

plot_after_response_m <- melt(plot_after_response, measure.vars = c('response', 'failure'))
plot_after_response_m$variable <- factor(plot_after_response_m$variable, levels = c('failure', 'response'))
plot_after_response_m$genus <- factor(plot_after_response_m$genus, levels = unique(plot_after_response$genus))

p1 <- ggplot(plot_after_response_m, aes(x = genus, y= sqrt(value/100000+0), fill=variable))
+ #, aes(x = genus), color = sex
  geom_bar(stat="identity", position="dodge",width = 0.6, alpha=1)+
  coord_flip()+
  labs(title="", x="", y="Abundance sqrt")+ #Marked genus in patients after FMT (q<0.05)
  theme(text=element_text(family="sans", size=32), plot.title = element_text(size=34, hjust = 0.5), axis.text = element_text(size=32, color='dimgray'), axis.title.x = element_text(size=34), axis.title.y = element_text(size=34), axis.ticks = element_blank())+
  theme(aspect.ratio=2, legend.position = c(4, .65), legend.background=element_rect(fill = NA), legend.text = element_text(size=0))+
  scale_fill_manual(values=c("#90908D", "#4D9127", "#7A1D1E", "#C47737", "#E7A600"))+
  # scale_alpha_manual(values = c(0.8))+
  theme(panel.background = element_rect(fill = NA, colour = "lightgrey", size = 3)
    ,axis.line=element_line(colour="lightgrey")
    ,axis.text.y = element_text(size=24, angle = 0, face = 'italic')
    # ,axis.text.x = element_text(size=21, angle = 0)
    ,axis.ticks = element_blank())+theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())

###weighted abundance of after response group
sign_response <- sign(tmp_pval_adjust05$gfc)
# print(tmp_pval_adjust05)

```

```

L6_rela_fil_sAg_remove_simp_weighted_response <- ifelse(sign_response>0, 1, 0) %%% tmp_L6_
05
L6_rela_fil_sAg_remove_simp_weighted_failure <- ifelse(sign_response<0, 1, 0) %%% tmp_L6_05

###
plot_marked_response_before <- function(L6_rela_fil_sAg_remove_simp_weighted_response, fla
g){
  tmp_3column_before <- meta_fil_config1_na[meta_fil_config1_na$pre_entro %in% c(pre_entr
o), c('Previous_sra', 'SRA_Sample', 'Donor_sra')]
  colnames(tmp_3column_before) <- c('Before', 'After', 'Donor')
  FMTstage_before <- melt(tmp_3column_before, measure.vars = c('Before', 'After', 'Donor'
))
  FMTstage_before$variable <- factor(FMTstage_before$variable, levels = c('Before', 'Afte
r', 'Donor'))

  FMTstage_abun_before <- unique(merge(FMTstage_before, t(rbind(colnames(L6_rela_fil_sAg_
remove_simp_weighted_response), L6_rela_fil_sAg_remove_simp_weighted_response)), by.x = 'value'
, by.y = 1))#unique(cbind(FMTstage_before, L6_rela_fil_sAg_remove_simp_weighted_response[,FMTst
age_before$value]))#
  colnames(FMTstage_abun_before) <- c('value', 'variable', 'abun')
  FMTstage_abun_before$abun <- as.numeric(as.character(FMTstage_abun_before$abun))
  ##plot
  # stat_dat <- FMTstage_abun_before %>% group_by(variable) %>% summarise(median = median
(log(abun+1)), q75 = quantile(log(abun+1), 0.75), q25=quantile(log(abun+1), 0.25))

  p<- ggplot(data = FMTstage_abun_before,aes(x = variable, y = log(abun + 1)))+ #FMTstage
_abun_before, x = 'variable', y = 'log(abun+1)', fill='NA', color='NA'
  # geom_violin(aes(x = variable, y = log(abun + 0), fill = variable), FMTstage_abun_be
fore, width=0.8)+
  geom_quasirandom(aes( color = variable), width = 0.3, alpha=0.8, FMTstage_abun_befor
e, method='smiley', size=2)+#color = ifelse(flag, "#4D9127", "#90908D"),
  stat_summary(mapping = aes(x = variable, y = log(abun + 1)), color='gray30', fun.min
= median, fun.max = median, fun = median, geom = "crossbar", width = 0.5, size=.6, alpha=1)+#,
color=variable
  stat_summary(mapping = aes(x = variable, y = log(abun + 1)), color='gray30', fun.min
= function(z) { quantile(z,0.75) }, fun.max = function(z) { quantile(z,0.75) }, fun = function
(z) { quantile(z,0.75) }, geom = "crossbar", width = 0.3, size=.4, alpha=1)+
  stat_summary(mapping = aes(x = variable, y = log(abun + 1)), color='gray30', fun.min
= function(z) { quantile(z,0.25) }, fun.max = function(z) { quantile(z,0.25) }, fun = function
(z) { quantile(z,0.25) }, geom = "crossbar", width = 0.3, size=.4, alpha=1)+
  scale_y_continuous(expand = c(0, 1))+
  stat_compare_means(comparisons = list(c('Before', 'After'), c('After', 'Donor'), c('B
efore', 'Donor')), method = 'wilcox.test', label = "p.signif", size=12)+
  labs(x= c(''), y=c(ifelse(flag,'Cumulative abundance of\nresponse-enriched taxa', 'Cumulative abundance of\nresponse-depleted taxa')), title = c())+
  scale_color_manual(name="FMT", values=c(donor_before_after_color))+
  scale_fill_manual(name="FMT", values=c(donor_before_after_color, "#962E2B", "#4E86C6", "#
4D9127", "#90908D", 'lightgrey'))+#' #C77CFF', '#43AFC8',
  theme(text=element_text(family = "sans", size=32), plot.title = element_text(size=34,
hjust = 0.5), axis.text = element_text(size=32, color = 'dimgray'), axis.title.x = element_text
(size=34), axis.title.y = element_text(size=34), axis.ticks = element_blank(), axis.ticks.y = e
lement_line(size=1.5, color = 'dimgray'), axis.ticks.length = unit(7, "pt"))+
  theme(aspect.ratio = 0.95, legend.background=element_blank(), legend.position=c(4, 0.6)
, panel.background = element_rect(fill = NA, colour = "lightgrey", size = 3)
, axis.line=element_line(colour="lightgrey")
, legend.key = element_rect(fill = NA, color = NA))+
  guides(colour = guide_legend(override.aes = list(size=5)))+theme(panel.grid = element_b

```

```

lank())
}

p2 <- plot_marked_response_before(L6_rela_fil_sAg_remove_simp_weighted_response, 1)
p3 <- plot_marked_response_before(L6_rela_fil_sAg_remove_simp_weighted_failure, 0)
if(dim(p1$data)[1] > 0){list((p1), (p2), (p3))}
}

```

Hide

```

pre_entro <- c('before1', 'before2')

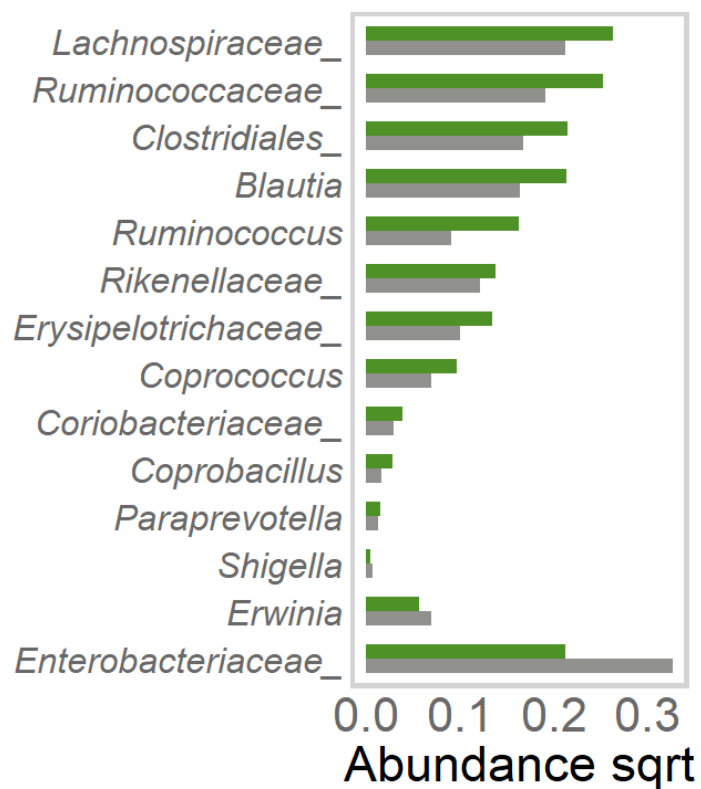
# pdf('figure2/2main_engraft_goog.pdf')
response_abundance(pre_entro)

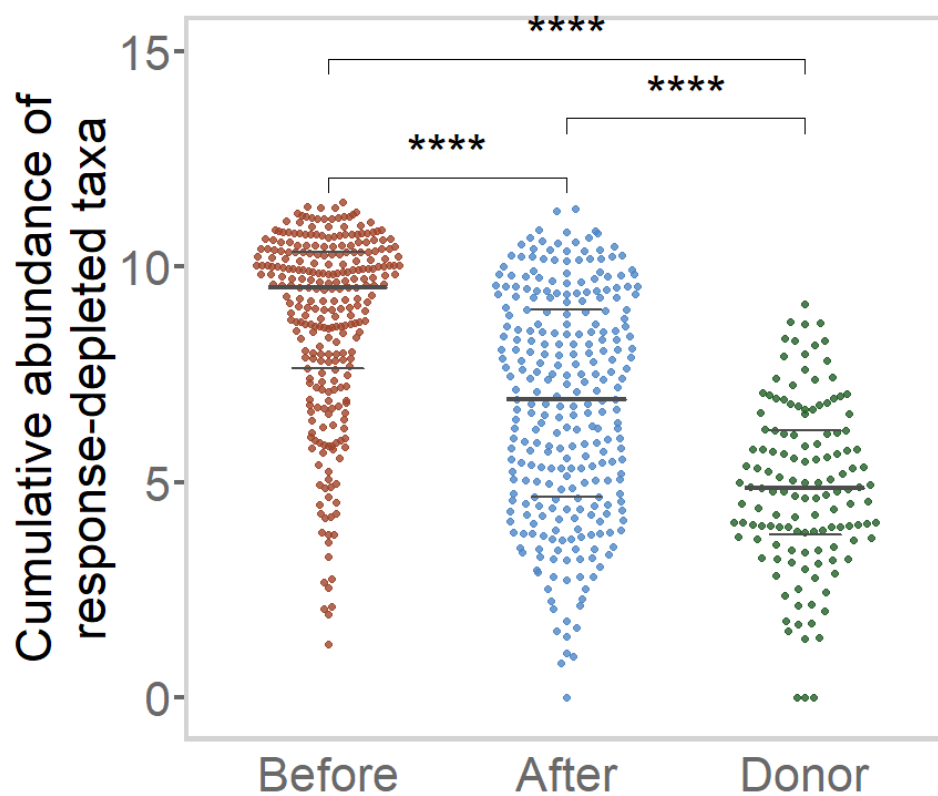
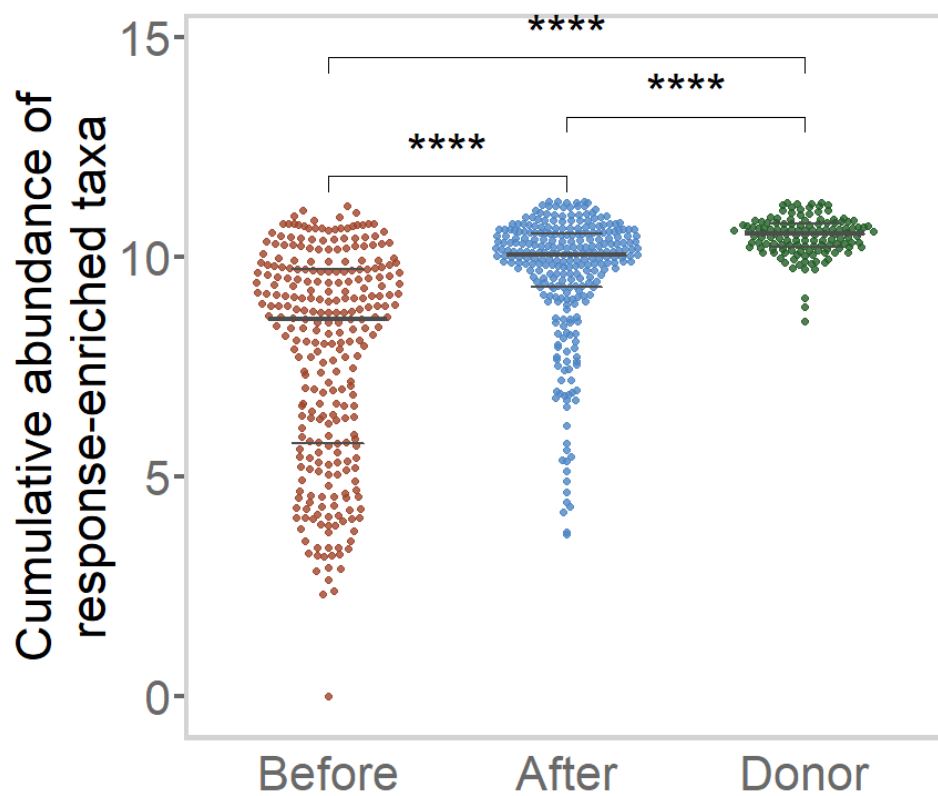
```

[[1]]

[[2]]

[[3]]





Hide

```
# dev.off()

# response_abundance('before1')
```

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Cmd+Option+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Cmd+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.